Diplomarbeit IRS-97-S-05

Entwicklung eines Werkzeugs für Entwurf und Simulation synergetisch vernetzter Subsysteme von Raumstationen

cand. aer. Jan Osburg, M. S.
Institut für Raumfahrtsysteme
Universität Stuttgart

Stuttgart, Mai 1997

Aufgabenstellung

Übersicht

Zur Erweiterung des am Institut für Raumfahrtsysteme vorhandenen Programmpaketes "Space Station Design Workshop" auf dem Gebiet der Modellierung und Simulation verknüpfter Systeme – demonstriert am Beispiel des Lebenserhaltungssystems (ECLSS) – wurde die Simulationsumgebung MELISSA entwickelt. Die neue Software basiert auf dem kommerziellen Programm LabVIEW, welches eine komfortable graphische Programmiersprache und Benutzeroberfläche zur Verfügung stellt.

Mögliche Einsatzbereiche von MELISSA reichen vom Analysieren und Validieren neuer ECLSS-Konzepte über das Optimieren von Parametern existierender Systeme bis zur Untersuchung der Auswirkungen von Teilausfällen und Subsystemfehlern auf das Gesamt-Lebenserhaltungssystem. Zusätzliche Anwendungsgebiete ergeben sich aus dem Einsatz beim Training von ECLSS-Entwicklern und -Anwendern, sowie im Bereich der universitären Ausbildung als Lehrwerkzeug in Seminaren zum Systementwurf.

Als Nachweis der vielseitigen Fähigkeiten von MELISSA wurde das ECLSS der Internationalen Raumstation modelliert und sein dynamisches Verhalten, sowohl im Normalbetrieb als auch bei Subsystemausfall, analysiert. Auch im Rahmen eines internationalen Raumstations-Workshops wurde MELISSA bereits angewendet.

Abstract

A new simulation environment, MELISSA, has been added to the "Space Station Design Workshop" software package existing at the Institut für Raumfahrtsysteme. The new environment supports the modeling and simulation of network-type subsystems like the Environmental Control and Life Support System (ECLSS). It is based on the commercial software LabVIEW, which provides a convenient graphical programming environment and user interface.

Applications of MELISSA range from the analysis and validation of new ECLSS designs to parametric optimization studies to failure mode effects and criticality analysis of a life support system. Additionally, MELISSA can be employed for training ECLSS developers and users, and as a teaching tool for lectures and seminars on system design.

As a demonstration of the powerful capabilities of MELISSA, the ECLSS of the International Space Station was modeled. Its time-dependent behavior was analyzed under normal operation conditions and with partial subsystem failures. MELISSA was also used during an international workshop on space station design.

Schlüsselwörter / Keywords

Systemsimulation; Prozeßsimulation; Synergetische Simulation; Lebenserhaltungssysteme; LabVIEW; Raumstationsentwurf; Systementwurf; Internationale Raumstation

Systems simulation; process simulation; synergistic simulation; life support systems; LabVIEW; space station design; systems design; International Space Station

Vorwort

The Earth is just too small and fragile a basket for the human race to keep all its eggs in.
Robert A. Heinlein [21]

Diese Diplomarbeit entstand in einer Zeit, in der die mittel- und langfristigen Anwendungsaussichten des behandelten Themas “Lebenserhaltungssysteme” nicht sicher beurteilt werden können: Auf der einen Seite muß die bemannte Raumfahrt – wie jeder hauptsächlich von staatlichen Organisationen getragene Bereich der Großforschung in Europa – aufgrund der gesamtwirtschaftlichen Lage weiterhin mit knappen Mitteln leben. Andererseits bieten sich gerade in den nächsten Jahren durch die Indienststellung der Internationalen Raumstation nie zuvor dagewesene Möglichkeiten, um Technologien und Abläufe für eine dauerhafte Präsenz der Menschheit im All zu entwickeln.

Die Notwendigkeit dieser Präsenz wurde bereits an anderer Stelle ([23], [5], [22], [28]) überzeugend begründet, ebenso wie die Vorteile einer europäische Beteiligung an der Internationalen Raumstation [7]. Die vorliegende Arbeit beschäftigt sich daher mit der eher pragmatischen Fragestellung, wie die Entwurfsarbeit des Systemingenieurs im für bemannte Raumfahrtprojekte zentralen Bereich des Lebenserhaltungssystems sowie bei ähnlich strukturierten Entwurfsaufgaben unterstützt werden kann. Dazu wurde nach eingehenden Voruntersuchungen als konkreter Lösungsansatz das Simulationswerkzeug MELISSA entwickelt, das nachfolgend vorgestellt wird.

Die Entwicklungsarbeit wurde durch die Mitarbeiter der Arbeitsgruppe “Raumfahrttechnologie und Raumfahrtnutzung” am Institut für Raumfahrtsysteme sehr erleichtert, denen ich dafür meinen herzlichen Dank ausspreche. Besonders bedanken möchte ich mich bei meinem Betreuer, Reinhold Bertrand, sowie bei Arndt Hinüber und Jörg Krüger für ihre kontinuierliche Unterstützung.

An dieser Stelle möchte ich auch die großzügige Hilfestellung durch die Firma National Instruments hervorheben, die das MELISSA zugrundeliegende Programm LabVIEW bereitwillig kostenfrei zur Verfügung stellte.

Die vorliegende Arbeit ist in Dankbarkeit meinen Eltern gewidmet, auf deren liebevolle Begleitung ich mich auf dem langen Weg zum Diplom stets verlassen konnte.

Inhaltsverzeichnis

1. Einleitung

Raumstationen haben verglichen mit anderen Systemen der Raumfahrt einen deutlich höheren Nachschubbedarf: Nutzlasten müssen entsprechend den Wünschen der Anwender im Orbit installiert oder ausgetauscht werden, und der laufende Betrieb erfordert die Bereitstellung von Verbrauchsstoffen für die Lage- und Bahnregelung (*Attitude and Orbit Control System*, AOCS) sowie für das Lebenserhaltungssystem (*Environmental Control and Life Support System*, ECLSS).

Während die zu transportierende Nutzlastmenge durch die Mission vorgegeben ist und daher vom Entwurfsingenieur nur wenig beeinflußt werden kann, ist der Transportbedarf speziell für das Lebenserhaltungssystem in hohem Maße abhängig von der Auslegung des zugehörigen Subsystems, wie Abbildung 1.1 zeigt. Dadurch kommt der frühzeitigen Einbeziehung des ECLSS in den Systementwurfsprozeß besondere Bedeutung zu.

In der Anfangsphase ist die Entwurfsaufgabe dadurch gekennzeichnet, daß nur unscharfe Vorgaben und Ziele vorhanden sind. Gleichzeitig macht die iterative Vorgehensweise eine Automatisierung der einzelnen Entwurfsschritte empfehlenswert. Für den Bereich des Systementwurfs auf Architektur- und Konfigurationsebene stehen im Rahmen des Space Station Design Workshops (SSDW) bereits Entwurfswerkzeuge zur Verfügung: GISSAD zum computergestützten Konfigurationsdefinition und IRIS zur bahndynamischen und Mikrogravitationssimulation.

Die Untersuchung vernetzter Subsysteme – wie z. B. des ECLSS – ist mit diesen Programmen jedoch nicht möglich, da die zwischen deren Komponenten bestehenden vielfältigen synergetischen Kopplungen nicht modelliert werden können. Daher ist die Entwicklung eines dieser Aufgabenstellung angepaßten Entwurfs- und Simulationswerkzeugs erforderlich.

1.1. Entwurf und Simulation

Der erfolgreiche Entwurf resultiert in der Definition eines Systems, das eine vorgegebene Aufgabe erfüllt. Bei der Entwicklung von Raumfahrtsystemen führen jedoch die Komplexität des Themas und vielfältige Interaktionen zwischen Unterbereichen, wie sie z. B. in [17] oder [33] dargestellt sind, während dieses Prozesses oft zum Verlust der Anschaulichkeit, die eine Voraussetzung für effizientes ingenieurmäßiges Entwerfen ist.

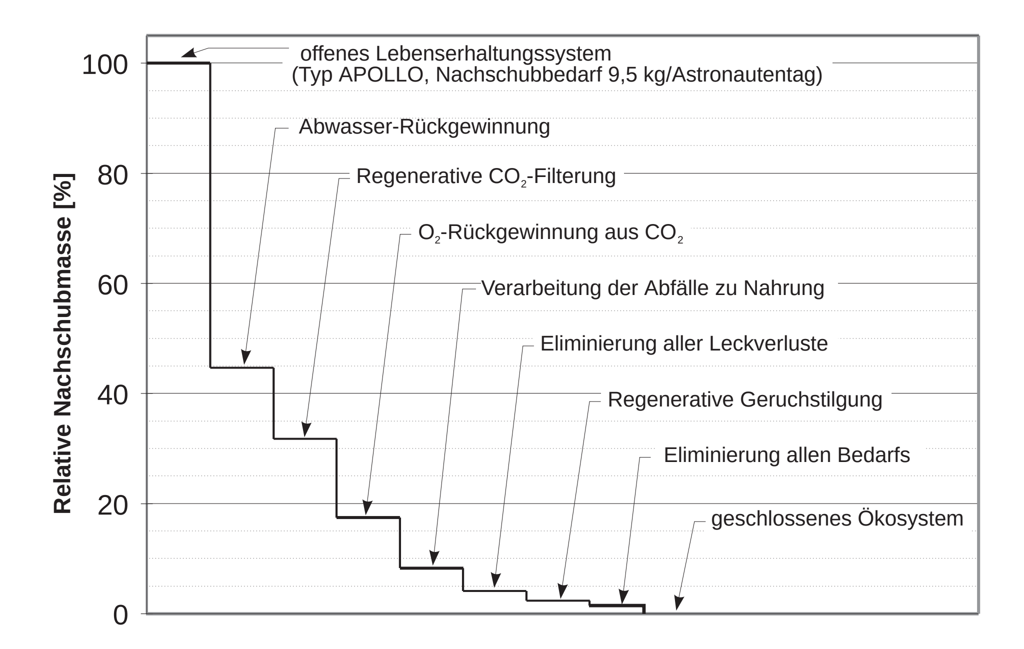


Abbildung 1.1.: Logistikbedarf verschiedener ECLS-Konzepte (aus [17])

Abhilfe schafft hier neben der Unterteilung der Entwurfsarbeit in überschaubare Teilaufgaben besonders die Verwendung von Simulationen zur Analyse des Systemverhaltens. Simulationen erlauben dem Entwurfsingenieur darüber hinaus, durch direkten – teilweise nur von der Intuition geleiteten – Umgang mit dem modellierten System vertiefte Einblicke in dessen dynamisches Verhalten zu gewinnen und so die Konvergenzrate des Entwurfsprozesses zu erhöhen.

Hier erkennt man auch den engen Zusammenhang mit einer Anwendung von Simulationssoftware im Rahmen der Lehre: Die Ausbildung im Systementwurf muß die oben beschriebene Praxis aufnehmen und dadurch über eine rein theoretische Vermittlung von Wissen hinausgehen. Das Konzept des *Learning by doing* hat sich als äußerst wirkungsvoll erwiesen, wie beispielsweise George Pólya feststellt ([31], S. 429):

> Learning should be active, not merely passive or receptive; merely by reading books or listening to lectures or looking at moving pictures without adding some action of your own mind you can hardly learn anything and certainly you cannot learn much.

Dieses "aktive" Lernen ist in vielen Bereichen der Technik, so auch in der Raumfahrt, nur unter Verwendung von Simulationen möglich. Der direkte Umgang mit den interessierenden Systemen kommt aufgrund des hohen finanziellen und zeitlichen Aufwandes sowie wegen meist stark beschränkter oder nicht vorhandener Verfügbarkeit entsprechender Hardware nicht in Betracht.

Dabei muß die Simulation nicht ein perfektes Bild der Wirklichkeit bieten, und in den meisten Fällen ließe sich das auch nicht erreichen. Simulation verfolgt immer einen bestimmten Zweck; wenn dieser erfüllt werden kann, sind die Modellierung und die darauf aufbauenden Berechnungen ausreichend. Marvin Minsky ([26], S. 7) liefert hierzu die pragmatische Definition:

> To an observer B an object M is a model of an object A to the extent that B can use M to answer questions that interest him about A.

Dementsprechend wird auch bei der im Rahmen dieser Diplomarbeit erstellten Simulationsumgebung auf einen gesunden Mittelweg zwischen Detailtreue auf der einen und Programmier- und Bedienungsaufwand auf der anderen Seite Wert gelegt.

1.2. Lebenserhaltungssysteme

Lebenserhaltungssysteme sind eine unabdingbare Voraussetzung bemannter Raumfahrtmissionen [34]. Auch in anderen Bereichen werden sie eingesetzt, um Menschen einen längeren Aufenthalt in lebensfeindlicher Umgebung zu ermöglichen, so z. B. in Unterseebooten oder Schutzräumen ([5], S. 425).

Der derzeitige Stand der Technik erlaubt jedoch nur relativ einfache Systeme. Es dominieren Konzepte, die auf externem Nachschub von Verbrauchsgütern basieren und meist nur einfache Luft- und Wasserreinigungsprozesse anwenden ([5], S. 134).

Für zukünftig geplante Raumfahrtprojekte wie Mond- oder Marsmissionen, oder auch zum Ausbau der im Entstehen begriffenen Internationalen Raumstation, müssen jedoch diese Kreisläufe wirksamer geschlossen werden [34]. Dabei werden fortgeschrittene physikalisch-chemische Verfahren zur Anwendung kommen, deren komplexes Zusammenwirken den Einsatz von Simulationssoftware wie des im Rahmen dieser Arbeit entwickelten Programmes MELISSA[1] beim Entwurf erforderlich macht.

Die vollständige Schließung aller lebensnotwendigen Stoffkreisläufe kann aber nur unter Einbeziehung biologischer Komponenten erreicht werden, wie sie ansatzweise bei diversen Biosphärenprojekten versucht wurde ([5], S. 249). Die hierzu notwendigen Elemente sind bisher noch nicht ausreichend erforscht, so daß auf ihre Modellierung im Rahmen dieser Arbeit verzichtet wird.

[1] *M*odular *E*nvironment for *Li*fe-Support *S*ystems *S*imulation and *A*nalysis

1.3. Gliederung des Berichts

An dieser Stelle soll ein kurzer Überblick über die Gliederung dieses Berichtes gegeben werden. Die Kapiteleinteilung richtet sich nach den ausgeführten Arbeitsschritten.

Simulationsumgebung

Zunächst wird in Kapitel 2 auf die Grundlagen der Erstellung der Simulationsumgebung eingegangen. Dazu gehört die Herleitung der Anforderungen sowie die Auswahl des Basiswerkzeuges. Darüberhinaus wird das hierzu ausgewählte Programm genauer beschrieben. Das darauffolgende Kapitel stellt dann die entwickelte Simulationsumgebung MELISSA im Detail vor, von den Grundlagen der Simulationserstellung bis zur Dokumentation der entwickelten ECLS-Modulbibliotheken.

Anwendungen

Mit einer ersten Beispielanwendung im Rahmen von Untersuchungen zur Internationalen Raumstation befaßt sich Kapitel 4. Das Lebenserhaltungssystem des amerikanischen Teils der ISS wurde modelliert und sein Verhalten unter verschiedenen Betriebsbedingungen simuliert. In Kapitel 5 wird der erste Einsatz der Simulationsumgebung im Rahmen der Lehre beschrieben. Hierbei wurde MELISSA den Teilnehmern des Space Station Design Workshops zur Verfügung gestellt, um die ECLS-Subsysteme der im Rahmen dieser Veranstaltung entworfenen Raumstationen zu simulieren.

Zusammenfassung

Schließlich werden im letzten Textkapitel die wichtigsten Punkte noch einmal zusammengefaßt und diskutiert sowie ein Ausblick auf weiterführende Arbeiten gegeben. Ein ausführliches Literaturverzeichnis schließt den Hauptteil dieses Berichtes ab.

Anhang

Im Anhang sind neben einem Begriffsverzeichnis zunächst die vollständigen Ergebnisse der Simulationen zu Kapitel 4 enthalten. Danach werden alle Bestandteile von MELISSA sowie die modellierten Beispiel-Lebenserhaltungssysteme durch ihre kommentierten Quellcodeausdrucke umfassend dokumentiert. Die ebenfalls im Rahmen dieser Diplomarbeit verfaßten englischsprachigen Unterlagen zur Simulationsumgebung – ein Benutzerhandbuch und eine Kurzreferenz – schließen diesen Bericht ab.

2. Grundlagen

Die zielgerichtete und effiziente Implementierung der Entwurfs- und Simulationsumgebung erfordert eine detaillierte Kenntnis der spezifischen Anforderungen, die der potentielle Benutzer an das Werkzeug stellt. Dadurch werden zugleich die Kriterien formuliert, welche die Auswahl des der Umgebung zugrundeliegenden Basisprogramms bestimmen. In den folgenden Abschnitten werden diese Anforderungen sowie der Auswahlprozeß dokumentiert und das gewählte Basiswerkzeug vorgestellt.

2.1. Anforderungen an die Entwurfs- und Simulationsumgebung

Bereits in der Aufgabenstellung werden mit den Begriffen "transparent" und "ergonomisch" grundlegende Leistungsmerkmale der Simulationsumgebung spezifiziert. Diese Forderungen bedürfen jedoch noch einer genaueren Definition, um praktisch umgesetzt werden zu können. Auch in Bezug auf die geforderte "graphisch-interaktive" Benutzeroberfläche sind genauere Festlegungen von Nutzen.

2.1.1. Erwartungen der Benutzer

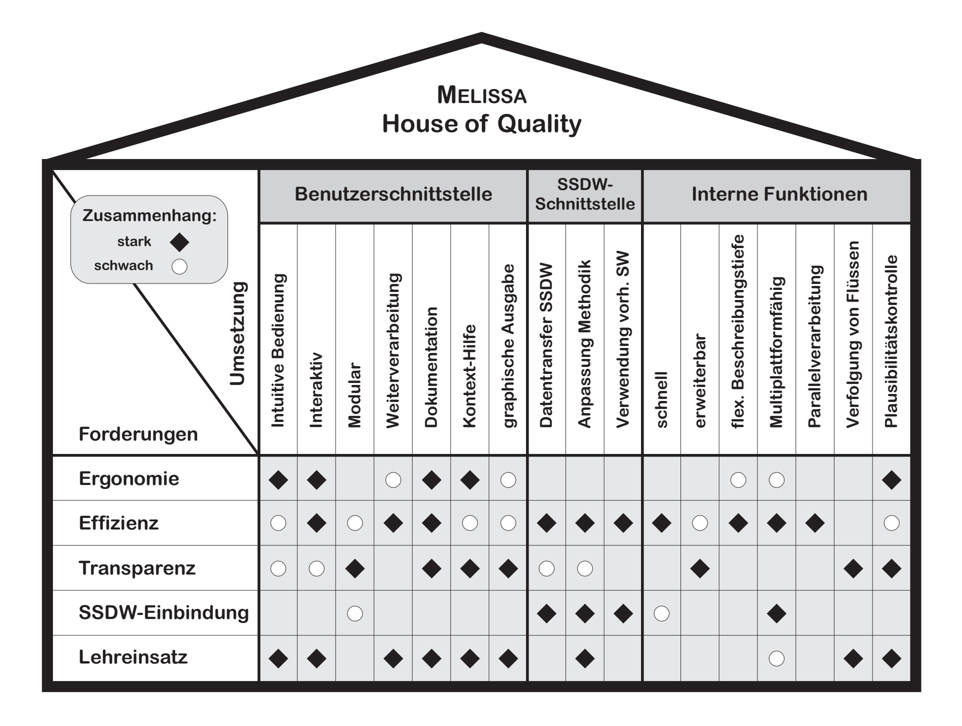
Die Simulationsumgebung soll sowohl den Systemingenieur bei Entwurf und Analyse des Lebenserhaltungssystems und anderer Subsysteme unterstützen als auch in anschaulicher Weise Seminarteilnehmern die Zusammenhänge komplexer Systeme wie des Lebenserhaltungssystems nahebringen.

Kundenorientierung

Damit die Software diesen Ansprüchen gerecht wird, können die von der Aufgabenstellung geforderten allgemeinen Eigenschaften im Rahmen des kundenorientierten Entwurfsprozesses mittels einer *House of Quality*-Matrix auf strukturierte und effiziente Weise in eine Liste konkreter Designvorgaben transformiert werden ([15], [29], [24], [8]).

Die Erstellung einer solchen Matrix (siehe Abbildung 2.1) erfordert folgende Schritte:

1. Schlagwortartiges Zusammenfassen der Kundenforderungen in der "Forderungen"-Spalte.



MELISSA
House of Quality

Zusammenhang: stark ◆ schwach ○

Forderungen \ Umsetzung	Benutzerschnittstelle							SSDW-Schnittstelle			Interne Funktionen						
	Intuitive Bedienung	Interaktiv	Modular	Weiterverarbeitung	Dokumentation	Kontext-Hilfe	graphische Ausgabe	Datentransfer SSDW	Anpassung Methodik	Verwendung vorh. SW	schnell	erweiterbar	flex. Beschreibungstiefe	Multiplattformfähig	Parallelverarbeitung	Verfolgung von Flüssen	Plausibilitätskontrolle
Ergonomie	◆	◆		○	◆	◆	○						○	○			◆
Effizienz	○	◆	○	◆	◆	○	○	◆	◆	◆	◆	○	◆	◆	◆		○
Transparenz	○	○	◆		◆	◆	◆	○	○			◆				◆	◆
SSDW-Einbindung			○					◆	◆	◆	○			◆			
Lehreinsatz	◆	◆		◆	◆	◆	◆		◆					○		◆	◆

Abbildung 2.1.: Die ausgefüllte *House of Quality*-Matrix

2. Assoziatives Finden von umsetzbaren Programmeigenschaften, die diese Forderungen unterstützen.
3. Ordnen dieser Eigenschaften nach Oberbegriffen in der “Umsetzung”-Zeile.
4. Dokumentieren des Zusammenhanges zwischen Kundenforderungen und Umsetzungsaspekten in der Matrix.
5. Gegebenenfalls Entfernen von Umsetzungsaspekten, die sich nach dem vorangegangenen Schritt als nicht relevant erweisen (d. h. keine Zusammenhangssymbole in ihrer Spalte haben).
6. Gegebenenfalls Hinzufügen weiterer Aspekte, welche die Umsetzung von zuvor unterrepräsentierten Kundenforderungen sicherstellen.

Durch diese Vorgehensweise wird gewährleistet, daß den allgemein gehaltenen Kundenforderungen direkt umsetzbare und objektiv überprüfbare Entwurfsvorgaben gegenüberstehen. Abbildung 2.1 zeigt die fertig ausgefüllte Matrix für den im Rahmen dieser Diplomarbeit durchgeführten Entwurfsprozeß. Im abschließenden Kapitel 6 wird, basierend auf der hier erzeugten Matrix, untersucht, inwieweit das entwickelte Programm die anfänglichen Erwartungen erfüllt.

Die durch dieses Verfahren gewonnene Anforderungsliste repräsentiert die Wünsche des “Kunden” und bestimmt so die weitere Entwicklungsarbeit:

- Schnittstelle zum Benutzer
  - intuitive Bedienung
  - interaktives Erstellen und Ausführen von Simulationen
  - objektorientierter modularer Aufbau[1]
  - automatisierter Datentransfer zur bequemen Weiterverarbeitung
  - ausführliche Dokumentation
  - kontextsensitive Hilfefunktion
  - graphische Ausgabe von Simulationsergebnissen
- Schnittstelle zur existierenden Software (SSDW)
  - automatisierter Datentransfer
  - Einpassung in Entwurfsmethodik
  - Benutzung vorhandener SSDW-Software für bestimmte Aufgaben
- Interne Funktionen
  - hohe Ausführungsgeschwindigkeit
  - einfache Erweiterung durch neue Module
  - Unterstützung unterschiedlicher Beschreibungstiefe einzelner Bausteine
  - einfache Änderung der Beschreibungstiefe existierender Bausteine
  - Einsetzbarkeit auf mehreren Computerplattformen
  - parallele Ausführbarkeit des Codes bei entsprechender Hardware
  - Verfolgen der Systemmasse, sowie der Massen- und Energieströme im System; eventuell auch Verfolgen der Kosten, der Zuverlässigkeit und des Entwicklungsrisikos
  - Plausibilitätskontrolle für erstellte Systemmodelle

2.1.2. Entwurfsgrundsätze einer interaktiven Benutzerschnittstelle

Die Eigenschaften der geforderten "graphisch-interaktiven" Benutzerschnittstelle können über das durch die *House of Quality*-Methodik erreichte Maß hinaus noch weiter spezifiziert werden. Dem kommt in diesem Fall besondere Bedeutung zu, da aufgrund der nachfolgend in Abschnitt 2.2 erläuterten Auswahl der Basissoftware nicht nur die

[1] Dieser Punkt wird auch von erfahrenen Simulationsanwendern z. B. aus der Verfahrenstechnik unterstützt: "Um die systemtechnische Modellentwicklung zu vereinfachen und zu unterstützen, ist es angebracht, einen Satz von möglichst universellen Modellbausteinen für die Beschreibung von Teilsystemen und deren Verknüpfungen zu definieren, aus denen sich durch Aggregation eine Vielzahl der unterschiedlichsten verfahrenstechnischen Prozeßmodelle aufbauen läßt." ([26], S. 14)

eigentliche Simulations-Schnittstelle, sondern auch die vom Benutzer zur Simulations- und Modul*erstellung* verwendete Umgebung rein graphischer Natur ist.

Folgende Grundsätze werden von der herangezogenen Referenzliteratur [20] als wichtig herausgestellt:

Konsistenz: Software soll nachvollziehbare Funktionalität bieten und ein einheitliches Erscheinungsbild haben. Sowohl der ursprüngliche Autor als auch nachfolgende Entwickler müssen dabei Konsequenz zeigen. Diesem Zweck dienen die im MELISSA User Manual (Anhang E, ab Seite 247) zusammengestellten Design-Regeln.

Analogien: Durch Ähnlichkeiten mit dem Benutzer vertrauten Elementen und Vorgängen soll die Einarbeitung in die Software erleichtert und das Verständnis gefördert werden. Der Benutzer kann jedoch leicht verwirrt werden durch kleinere Abweichungen zwischen tatsächlichem Verhalten der Software und dem durch die Analogie nahegelegten. Daher wird hier besonderer Wert darauf gelegt, diese Abweichungen eingehend zu erläutern und auf Besonderheiten hinzuweisen (siehe Abschnitt 3.2.4).

Erfahrungswerte: Im Bereich der Gestaltung von graphischen Benutzeroberflächen gibt es viele Erfahrungswerte:

- Bevorzugung eines neutralen (hellgrauen) Hintergrundes
- Verwendung von höchstens fünf verschiedenen Farben gleichzeitig
- Vermeidung von Farben ähnlichen Helligkeits- oder Farbwerts
- Deutliche Kennzeichnung verwechselbarer Elemente
- Ausreichende Größe der Bedienelemente
- Einheitlicher und ausreichend großer Zeichensatz
- ...

Diese Grundsätze sind nicht nur bei der Aufstellung der oben erwähnten Design-Regeln zu berücksichtigen, sondern sie beeinflussen auch die nachfolgend beschriebene Auswahl des Basisprogrammes.

2.2. Auswahl des Basiswerkzeuges

Nachdem im vorangegangenen Abschnitt das Anforderungsprofil dokumentiert wurde, wird nun die Auswahl der Basis-Software beschrieben, auf der die eigentliche MELISSA-Simulationsumgebung basiert.

2.2.1. Übersicht über die zur Auswahl stehenden Werkzeuge

2.2.1.1. Verfahrenstechnische Software

Da die behandelte Thematik "Lebenserhaltungssysteme" sich hauptsächlich mit chemisch-physikalischen Prozessen befaßt, liegt es nahe, auch Programmpakete aus dem Bereich der Prozeßsimulation auf ihre Eignung hin zu untersuchen. Näher betrachtet werden sollen hier die kommerzielle Software "AspenPlus" [1] der amerikanischen Firma Aspen Technologies sowie der an der Universität Stuttgart entwickelte Prozeßsimulator "Diva" [26].

Beide Programme zeichnen sich durch detailgetreue Modellierung chemischer und physikalischer Prozesse aus, wobei zur Lösung der entstehenden Differentialgleichungssysteme Methoden der numerischen Mathematik eingesetzt werden. Jedoch ist auch eine entsprechend genaue Detailkenntnis der Prozesse in zu modellierenden Modulen notwendig, die normalerweise im frühen Stadium des Systementwurfs oder bei noch nicht fertigentwickelten Unterbaugruppen nicht vorhanden ist. Darüber hinaus benötigt ein Benutzer zur erfolgreichen Definition eigener Modelle erhebliche Erfahrung [14]. AspenPlus liefert außerdem nur stationäre Lösungen, eine dynamische Simulation ist erst mit weiterer Software (Erweiterungsprogramm "Speedup") möglich.

Detailgetreu

Erfahrung nötig

Die Programme bieten eine graphische Benutzeroberfläche, die sich aber auf Netzwerkeditoren zur Systemerstellung beschränkt; ansonsten ist die Benutzerschnittstelle textorientiert. Zur Steuerung und Erweiterung bedienen sich beide der Sprache FORTRAN [14]. Zudem orientieren sie sich an den Bedürfnissen der Großchemie, so daß keine ECLSS-spezifischen Komponenten vorhanden sind.

2.2.1.2. Mathematisch orientierte Software

Aus dem Bereich der Regelungstechnik bietet die Kombination Matlab/Simulink [27] Anlaß zur näheren Betrachtung. Ihr Hauptvorteil ist die durchgehend verwendete Zustandsraumdarstellung simulierter Systeme, die eine zeitkontinuierliche Berechnung ermöglicht. Auch hier erleichtert eine graphische Oberfläche die Bedienung, jedoch muß zur Erstellung benutzerdefinierter Module wieder auf den Textmodus zurückgegriffen werden.

Zustandsraum

Durch das Zustandsraumkonzept bedingt, müssen sämtliche Systemgrößen ständig zur Verfügung stehen, d. h. am Eingang jedes Bauteils muß der komplette Zustandsvektor vorliegen. Dies macht die Modellierung komplexer Systeme fehleranfällig und umständlich, zumal keine wirksamen Dokumentationsfunktionen zur Verfügung stehen. Auch die graphische Darstellung des Systemmodells ist eher einfach gehalten und praktisch ohne Anpassungsmöglichkeiten.

Simulink zielt wie Matlab auf Anwender aus der Regelungstechnik und Mathematik, so daß ebenfalls keine ECLSS-spezifischen Komponenten vorhanden sind.

2.2.1.3. Graphische Programmiersprachen

Auf einen völlig anderen Anwenderkreis zielt das kommerzielle Programm LabView [12]. Im Vordergrund steht bei diesem Produkt die Erfassung von Meßdaten und deren ansprechende Darstellung. Zu diesem Zweck bietet die Software eine mächtige graphische Programmiersprache, G, sowie zugehörige Ein- und Ausgabe-Elemente im Stil von dem Ingenieur vertrauten Meßgeräte-Bedienelementen und -Anzeigen. Zusätzliche Funktionen werden in umfangreichen Bibliotheken zur Verfügung gestellt. Die graphische Programmierumgebung ermöglicht dabei durch interne Compilierung eine Ausführungsgeschwindigkeit, die mit der von C-Code erreichbaren verglichen werden kann.

Meßdatenerfassung

Bibliotheken

Diese Bestandteile lassen sich für eine graphische Simulationsumgebung verwenden, jedoch müssen alle Elemente, die die Simulation und das Lebenserhaltungssystem betreffen, eigens programmiert werden. Die Software stellt andererseits viele Hilfe- und Dokumentationsfunktionen zur Verfügung. Auch das modulare, hierarchische Programmierkonzept ist von Vorteil. Die gesamte Umgebung kann dazu in vielen Bereichen an den Benutzer angepaßt werden.

Unspezifisch

Weitere Meßwerterfassungsprogramme mit graphischer Programmiersprache sollen hier aufgrund ihrer beschränkten allgemeinen Verwendbarkeit (siehe [30]) nicht berücksichtigt werden.

2.2.1.4. Vorhandene ECLSS-Simulationssoftware

Neben den oben vorgestellten, für den allgemeinen Anwendungsfall entworfenen Programmen gibt es bereits ECLSS-spezifische Software. Hier sind die von der NASA (MSFC) entwickelte Software "CASE/A" [2] sowie deren ESA-Pendant "Ecosim" [3] zu erwähnen. Beide Progamme bieten eine graphische Oberfläche für die Systemmodellierung, sowie Bibliotheken mit Lebenserhaltungssystem-spezifischen Komponenten. Die zugrundeliegenden Komponenten werden mit Differentialgleichungssystemen beschrieben.

ECLSS-spezifisch

Zur Erweiterung ist der Benutzer auf Texteingabe angewiesen. CASE/A benutzt dabei FORTRAN-ähnlichen Code. Im Falle von Ecosim muß der Benutzer auf C zurückgreifen. Bei beiden Programmen ist aufgrund des speziellen Anwendungszwecks mit erheblichen Einarbeitungszeiten zu rechnen. Weitere Nachteile der beschriebenen Programme sind ihre begrenzte Portabilität: CASE/A steht nur für DEC VAX Maschinen zur Verfügung, Ecosim nur für HP oder Sun Workstations. Darüberhinaus schließt auch der von der NASA festgesetzte Preis von US$ 12.000,- für internationale Lizenzen eine Verwendung von CASE/A im nichtkommerziellen Bereich aus.

2.2.2. Entscheidungskriterien

Ein Vergleich der Vor- und Nachteile aller oben vorgestellten Programme führt zu einer Entscheidung zugunsten von LabVIEW. Folgende Überlegungen tragen dazu bei:

- LabVIEW bietet eine umfassende und doch einfach zu benutzende graphische Programmierumgebung.
- Umfangreiche Funktionsbibliotheken erleichtern rasches und effizientes Programmieren ohne Beschränkung auf ECLSS-spezifische Fragestellungen.
- Erweiterungen sind aufgrund der hierarchischen und modularen Programmierung einfach möglich.
- Im Vergleich zu den oben vorgestellten Konkurrenten läßt sich mit LabVIEW das Konzept der graphischen Modellierung und Bedienung am konsequentesten implementieren.
- Durch die intuitive Bedienoberfläche werden neue Benutzer schnell mit dem Programm und darauf basierenden Anwendungen wie z. B. MELISSA vertraut.
- Die Bedienoberfläche ist bereits entsprechend den oben in Abschnitt 2.1.2 erwähnten Grundlagen gestaltet (siehe auch [10]).
- LabVIEW kann sowohl auf PCs (unter Windows 3.1, Windows 95 und Windows NT) als auch auf Macintosh- und diversen Unix-Plattformen eingesetzt werden.
- Der interne Compiler liefert schnellen Code, der bei entsprechender Hardware auch zur Parallelverarbeitung geeignet ist.
- Durch vorhandene Hardware-Interface-Funktionen kann gegebenenfalls auch "echte" Versuchshardware in Simulationsläufe eingebunden werden.

2.3. Kurze Beschreibung des ausgewählten Programms

Die folgenden Darstellungen sollen soweit mit LabVIEW vertraut machen, wie es zum Verständnis der weiteren Kapitel notwendig ist. Für eine weitergehende Einführung sei auf das MELISSA-Benutzerhandbuch im Anhang (ab Seite 247) verwiesen. Auch direkter Umgang mit LabVIEW stellt eine effiziente Ergänzung dar.

2.3.1. Programmkonzept

Ein LabVIEW-Programm, das aufgrund der meßtechnischen Ursprünge der Software als *Virtuelles Instrument* (VI)bezeichnet wird, besteht aus zwei Hauptelementen: ein *Front Panel* genanntes Fenster mit grauem Hintergrund sowie ein als *Diagramm* bezeichnetes

mit weißem Hintergrund. Diese übernehmen die Funktionen der von anderen Programmiersprachen her bekannten Elemente: Das Front Panel stellt die Schnittstelle eines LabVIEW-Programms zum Benutzer und zu anderen Programmen dar, auf dem Daten ein- und ausgegeben werden, während das Diagramm zum Implementieren des eigentlichen Algorithmus des VIs dient. Im Falle von LabVIEW wird hierzu die eingebaute graphische Programmiersprache G verwendet.

Front Panel

Diagramm

Ein *Palette* genanntes Menüfenster stellt die Werkzeuge zur Verfügung, durch die der Benutzer mit LabVIEW interagieren kann (Abbildung 2.2). Zur Erklärung der verschiedenen Elemente sei auf die LabVIEW-Kurzreferenz (Anhang F ab Seite 307) verwiesen.

"Tools" Palette

2.3.2. LabView Benutzerschnittstelle

Entsprechend dem graphischen und intuitiven Konzept von LabVIEW sind die zur Verfügung gestellten Schnittstellenelemente für das Front Panel so gestaltet, daß sie vertrauten Schaltern und Instrumenten ähneln: Drehknöpfe, Zeigerinstrumente, Kippschalter, Kurvenschreiber, und so fort. Diese Elemente werden durch einfaches Ziehen (*drag-and-drop*) von den graphischen Menüs, auf denen sie zur Verfügung stehen, auf das Front Panel übertragen. Das entsprechende Hauptmenü, die *Controls Palette*, ist in Abbildung 2.3 dargestellt. Ein Beispiel für ein Front Panel mit Instrumenten ist in Abbildung 2.5 gegeben.

Controls Palette

Front Panels werden im Rahmen von MELISSA zur Steuerung von Simulationsläufen und von Modulen des modellierten Systems verwendet, ferner zur Anzeige von Zwischen- und Endergebnissen. Das folgende Kapitel geht darauf noch näher ein. Auch das LabVIEW-Handbuch [12] sowie das zugehörige Tutorial und die Online-Hilfe stellen weitere Informationen über die hier angesprochenen Grundfunktionen der Benutzerschnittstelle zur Verfügung.

Handbuch

2.3.3. Graphisches Programmieren im Diagramm

Die Berechnungen und sonstigen Aktionen, die ein (Unter-)Programm ausführt, werden in dessen Diagrammfenster festgelegt. Wie schon auf dem Front Panel führt die graphische Natur von LabVIEW auch hier zu einer entsprechenden Vorgehensweise. Operatoren wie "+", Strukturen wie "While"-Schleifen und auch zusätzliche Unterprogramme und Module werden als Symbole (*icons*) eingefügt und durch "Drähte" (*wires*) verbunden, um den Datenfluß festzulegen. Dadurch werden zeichenbasierte Befehlssequenzen (z. B. `a := b + c`), wie sie aus anderen Programmiersprachen bekannt sind, überflüssig. Abbildung 2.6 zeigt, wie ein einfaches Beispielprogramm, das zwei Zahlen addiert, in LabVIEW aussehen könnte.

Wires

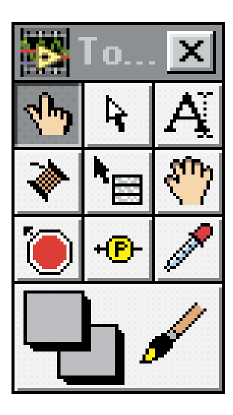


Abbildung 2.2.: LabView "Tools"-Palette

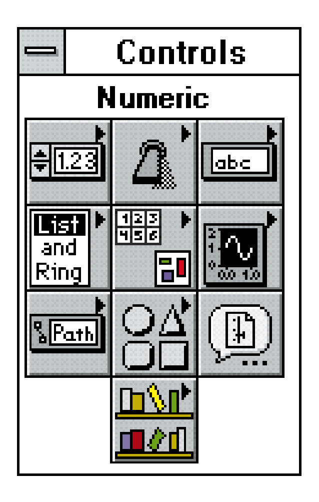


Abbildung 2.3.: "Controls" Palette

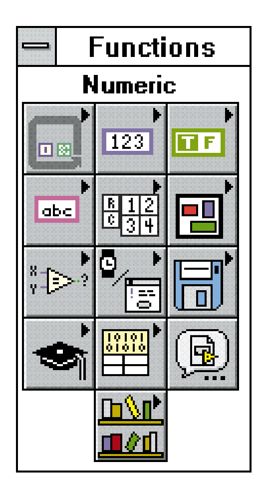


Abbildung 2.4.: "Functions" Palette

Functions Palette

Alle Elemente, die im Diagramm verwendet werden können, werden von einem weiteren Hauptmenü, der *Functions Palette*, zur Verfügung gestellt, die in Abbildung 2.4 dargestellt ist. Auch die im folgenden Kapitel vorgestellten MELISSA-Module sind über diese Palette erreichbar.

Bearbeitungsreihenfolge

Ein wichtiger Unterschied zwischen LabVIEW und herkömmlichen, textbasierten Programmiersprachen soll an dieser Stelle hervorgehoben werden: In LabVIEW hängt die Bearbeitungsreihenfolge nur vom Datenfluß zwischen den einzelnen Befehlen ab, die durch die "Verdrahtung" festgelegt wird. Keine Rolle spielt hingegen die räumliche Anordnung der Icons auf dem Diagramm oder die Reihenfolge, in der diese eingefügt wurden. Dies ist ein klarer Unterschied zu anderen Programmiersprachen, bei denen die Bearbeitungsreihenfolge von der (räumlichen) Anordnung der Befehle im Quelltext eines Programms abhängt. So wird zum Beispiel bei folgendem Programmcode

```
a      := read_input();
b      := read_input();
result := a  + b;
```

zunächst `a` eingelesen, danach `b`, bevor schließlich die Summe der beiden Zwischenspeicher an `result` zugewiesen wird.

Data flow programming

Mit LabVIEW wird dieser Algorithmus unter Zuhilfenahme von Symbolen und Drähten, ähnlich wie in Abbildung 2.6 dargestellt, ausgedrückt. Aufgrund des bei LabVIEW unterschiedlichen Abarbeitungskonzeptes, das als *data flow programming* bezeichnet wird, kann jedoch in diesem Fall nur gesagt werden, daß vor der Berechnung des Ergebnisses zunächst die Eingaben `a` und `b` abgefragt werden. Es ist jedoch nicht feststellbar, ob dabei `a` vor `b` eingelesen wird oder umgekehrt.

In diesem einfachen Beispiel ist dieser Unterschied aufgrund der Kommutativität des verwendeten Algorithmus natürlich bedeutungslos. Er muß jedoch bei komplexeren Programmieraufgaben stets berücksichtigt werden. Auch die Gestaltung von MELISSA-Simulationen und -Modulen wird dadurch beeinflußt. Dem Nachteil, daß einige Algorithmen mit expliziter Angabe der Ausführungsreihenfolge programmiert werden müssen

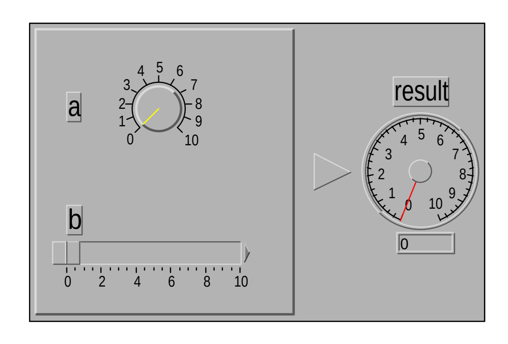


Abbildung 2.5.: Front Panel (Beispiel)

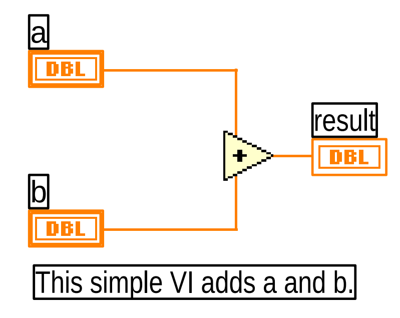


Abbildung 2.6.: Diagramm (Beispiel)

Vorteile

(z. B. die Zeitverwaltung oder Initialisierung), stehen jedoch deutliche Vorteile gegenüber. So kann die Anordnung der Programmelemente im Diagramm frei und nur nach Übersichtlichkeitskriterien erfolgen. Ferner wird die Ausführungsgeschwindigkeit des Codes erhöht, da die Bearbeitungsreihenfolge innerhalb der durch den Datenfluß gegebenen Grenzen vom Compiler optimiert werden kann. Auch die parallele Ausführbarkeit des Codes auf entsprechenden Rechnern wird gefördert.

2.3.4. LabView Programmausführung

Die Ausführung einer Melissa-Simulation kann nur aus LabView heraus stattfinden, da der kompilierte Code[2] nicht unabhängig lauffähig ist. Die dazu notwendigen Befehle



werden als Schaltflächen im LabView-*Toolbar* zur Verfügung gestellt (siehe nebenstehende Abbildung). Dies führt zu erhöhten Systemanforderungen für Melissa, da eine vollständige LabView-Installation vorhanden sein muß. Auf der anderen Seite hat der Benutzer dadurch immer die volle Leistungsfähigkeit des Basisprogrammes zur Verfügung.

2.3.5. Zusätzliche Funktionen

LabView stellt einige Funktionen zur Verfügung, die zum Programmieren nicht unmittelbar notwendig sind, die jedoch dazu beitragen, daß LabView die Anforderungen gemäß Abschnitt 2.1.1 erfüllt.

Fehlermeldung

Warnung

Wenn ein Programm aufgrund falschen Codes nicht ausgeführt werden kann, wird im Toolbar statt des Run-Schalters ein Fehlersymbol gezeigt, das nach Anwahl eine Liste mit allen von LabView gefunden Fehlern anzeigt. Auf Code, der zwar die Ausführung des Programmes nicht unmöglich macht, jedoch trotzdem Fehler enthalten könnte, wird durch ein Warnsymbol auf gleiche Weise aufmerksam gemacht.

Fehlersuche

Ein zusätzlicher Schalter steht zum Aufspüren von Laufzeitfehlern zur Verfügung. Wenn dieser gedrückt ist, wird das zugehörige VI im Zeitlupentempo ausgeführt, wobei alle Zwischenergebnisse auf dem Bildschirm angezeigt werden und der Datenfluß in den "Drähten" graphisch dargestellt wird.

Diese Funktionen helfen beim Erstellen korrekter Programme und reduzieren sowohl die anfängliche Einarbeitungszeit als auch den Programmieraufwand.

Dokumentation

Zum Ausdruck von VIs stellt LabView eine eingebaute Dokumentations-Ausdruckfunktion zur Verfügung. Mit dieser wurden die Melissa-Moduldokumentationen in Anhang C erzeugt. Weiterhin wird der Informationstext zu jeder VI in einem Online-Hilfefenster angezeigt. In diesem Informationstext ist auch eine Auflistung aller Ein- und Ausgänge des VIs enthalten. Besonders für unerfahrene oder gelegentliche Anwender von Melissa ist diese Funktion von Vorteil.

[2] Die Kompilierung wird von LabView automatisch beim Abspeichern durchgeführt.

3. Vorstellung der MELISSA-Simulationsumgebung

3.1. Allgemeiner Lösungsansatz

Die entwickelte Umgebung soll die Modellierung und Simulation von vernetzten Systemen ermöglichen. Da diese Systeme aus diskreten Komponenten aufgebaut sind und ihr Verhalten sich aus deren Interaktion ergibt, basiert auch die Modellierung auf der Verwendung vorbereiteter Unterprogramme, die einzelne Komponenten repräsentieren, und der Definition der Verbindungen zwischen diesen.

Komponenten

Die ECLSS- und EPS-Module, die in MELISSA-Simulationen verwendet werden können, sind als LabVIEW-Unterprogramme ("Virtuelle Instrumente", VIs) implementiert. Auf diese Weise werden die Eigenschaften der graphischen Oberfläche von LabVIEW genutzt. Auch die leichte Erweiterbarkeit von MELISSA durch Programmieren neuer Module oder Erhöhen der Beschreibungstiefe vorhandener Bausteine wird dadurch gewährleistet.

Algorithmen

Die VIs modellieren das Verhalten der entsprechenden Geräte durch die in ihnen enthaltenen Algorithmen. Dabei werden hauptsächlich arithmetische Beziehungen zwischen Ein- und Ausgangsgrößen verwendet, die – wo sinnvoll – durch vom Benutzer veränderbare Parameter angepaßt werden können. Abbildung 3.2 im nachfolgenden Abschnitt zeigt exemplarisch einen solchen arithmetischen Algorithmus in Form einen LabVIEW-Diagramms.

Auswahl

Die im Rahmen dieser Arbeit programmierten, dem Benutzer zur Verfügung stehenden Module simulieren typische Bestandteile des Lebenserhaltungs- und Energiesystems, wie Wasserfilter, Sauerstofftanks, Solarzellen oder auch Luft-Analysatoren. Es sind genügend Komponenten berücksichtigt, um ein einfaches, weitgehend geschlossenes Lebenserhaltungssystem zu modellieren. Weitere Module können bei Bedarf nach dem Beispiel der vorhandenen leicht erstellt werden (siehe auch Kapitel 6).

Simulation

Entsprechend dem graphischen Charakter von LabVIEW erfolgt auch die Simulationserstellung und -ausführung mit MELISSA durchweg auf graphische Weise. Das zu simulierende System wird zunächst unter Verwendung der vordefinierten Komponenten in schaltplanähnlicher Form am Bildschirm eingegeben. Daraufhin werden die gewünschten lokalen Anfangsbedingungen für die einzelnen Systemkomponenten eingestellt. Sobald ein Simulationslauf gestartet ist, kann der Benutzer das Systemverhalten am Bildschirm

Interaktive Simulation

verfolgen und durch Verändern von Parametern direkt beeinflussen. Durch diesen interaktiven Ansatz läßt sich das dynamische Verhalten eines modellierten Systems einfach und dennoch zuverlässig untersuchen. Selbstverständlich können auch zeitinvariante Größen wie z. B. die Systemmasse angezeigt werden.

Keine DGLs

Bei der dynamischen Simulation wird auf explizites Lösen von Differentialgleichungen, wie es z. B. in eher prozeßtechnisch orientierter Software wie AspenPlus [1] oder auch ECOSIM [3] angewandt wird, verzichtet. Statt dessen werden die im System zirkulierenden Stoff- und Energieströme direkt modelliert und bei Bedarf (z. B. in den Tank-Modulen) zeitdiskret integriert. Ein Simulationslauf besteht also aus aufeinanderfolgenden Simulationsschritten, von denen jeder einem Zeitinkrement von wählbarer Länge entspricht. Durch diesen Ansatz reduziert sich der Rechenaufwand für einen Simulationsschritt erheblich. Auch die Anschaulickeit der Modellierung sowohl von einzelnen ECLS-Komponenten als auch des Simulationsmodells insgesamt wird verbessert.

3.2. Konkrete Umsetzung: Simulationsumgebung MELISSA

Das Umsetzen des oben beschriebenen allgemeinen Ansatzes führte zur Erstellung der Entwurfs- und Simulationsumgebung MELISSA. In diesem Abschnitt werden Einzelheiten der Implementierung beschrieben und die programmierten Komponenten vorgestellt. Auch auf die Erstellung und Durchführung von Simulationen mit MELISSA sowie dabei auftretende Besonderheiten wird eingegangen.

3.2.1. Implementierung

Die Modellierung der Komponenten stützt sich hauptsächlich auf das Werk [5], jedoch entstammen einige Daten auch sonstigen Büchern ([17], [33], [28], [11], [4]) sowie verschiedenen Forschungsberichten ([6], [13], [35]).

Auslegung

Die meisten Komponenten des ECLSS oder EPS sind als konkrete Geräte bestimmter Größe oder Kapazität implementiert. Die Crewanzahl, für die diese ausgelegt sind, ist an der Zahl im VI-Namen erkennbar (z. B. ist das VCD_6.vi-Modul für sechs Personen ausgelegt). Einige Module sind jedoch auch skalierbar (erkennbar an einem angehängten "_S", z. B. Tank_H2O_s) oder besitzen eine unbeschränkte Kapazität (angehängtes "_X", z. B. CHX_x).

Ein- und Ausgänge

Die physikalisch notwendigen Ein- und Ausgänge sind als Front-Panel-Instrumente vorhanden. Darüber hinaus kann der Benutzer am Front Panel gegebenenfalls zusätzliche Parameter wie Wirkungsgrade oder Verarbeitungskapazitäten angeben. Die überwiegende Anzahl der VIs hat einen zusätzlichen Ausgang für die Gerätemasse, damit die Gesamtmasse eines Systems leicht ermittelt werden kann (wie z. B. bei dem in Abbildung 5.1 in Kapitel 5 dargestellten Modell). Bei skalierbaren Modulen wird die Gerätemasse intern entsprechend berechnet.

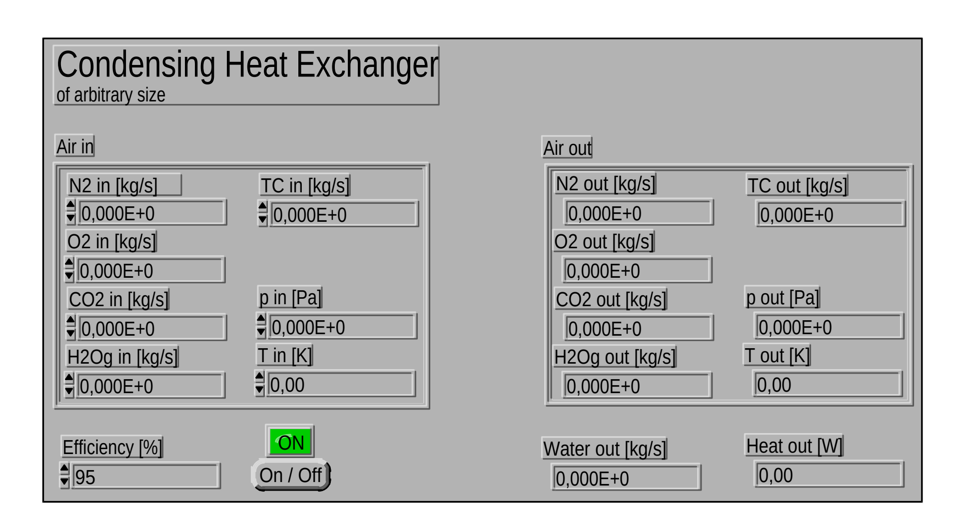


Abbildung 3.1.: Condensing Heat Exchanger Front Panel

Beispiel: CHX

Als Beispiel für eine einfache Implementierung dient der Luftentfeuchter/Wärmetauscher (*Condensing Heat Exchanger*, CHX), dessen Front Panel und Diagramm in Abbildung 3.1 bzw. 3.2 dargestellt sind. Die Ein- und Ausgänge dieses VIs sind ein Luftstrom sowie Kondenswasser und Wärme. Zusätzlich kann auf dem Front Panel noch der Wirkungsgrad des Geräts eingestellt sowie dieses an- oder ausgeschaltet werden.

Im Diagramm ist sichtbar, daß sich der Wasserdampfanteil des Luftstromes entsprechend dem gewählten Wirkungsgrad reduziert und proportional Kondensationswärme frei wird. Die Schaltlogik im unteren Teil des Diagrammes befindet sich in dieser Form in jedem schaltbaren VI. Sie sorgt für korrektes Verhalten der An- und Ausschalter auf dem Hauptsteuerfeld und auf den jeweiligen Geräte-Front-Panels. Die vier untersten rechteckigen Strukturen zeigen den alternativen Code für die IF-Abfragen.

Bibliotheken

Die erstellten Module sind entsprechend ihrem Verwendungszweck auf die folgenden drei Modulbibliotheken (*Libraries*) aufgeteilt:

ECLS Modules: Hier befinden sich alle VIs, die ECLSS- und EPS-spezifische Hardware simulieren. Eine Übersicht über die in dieser Bibliothek zusammengefaßten Komponenten bietet Abbildung 3.3.

ECLS Library: In dieser Bibliothek sind Module zusammengefaßt, die in einer Simulation ECLSS- und EPS-spezifische Hilfsfunktionen ausführen oder die Grundbausteine für die Module in der "ECLS Modules"-Library zur Verfügung stellen. Abbildung 3.4 zeigt die in dieser Bibliothek enthaltenen VIs.

Auxiliary Library: Einige allgemeine Funktionen ohne direkten ECLSS- oder EPS-Bezug, die während der Erstellung von MELISSA programmiert wurden, werden in dieser Library zur Verfügung gestellt (Abbildung 3.5).

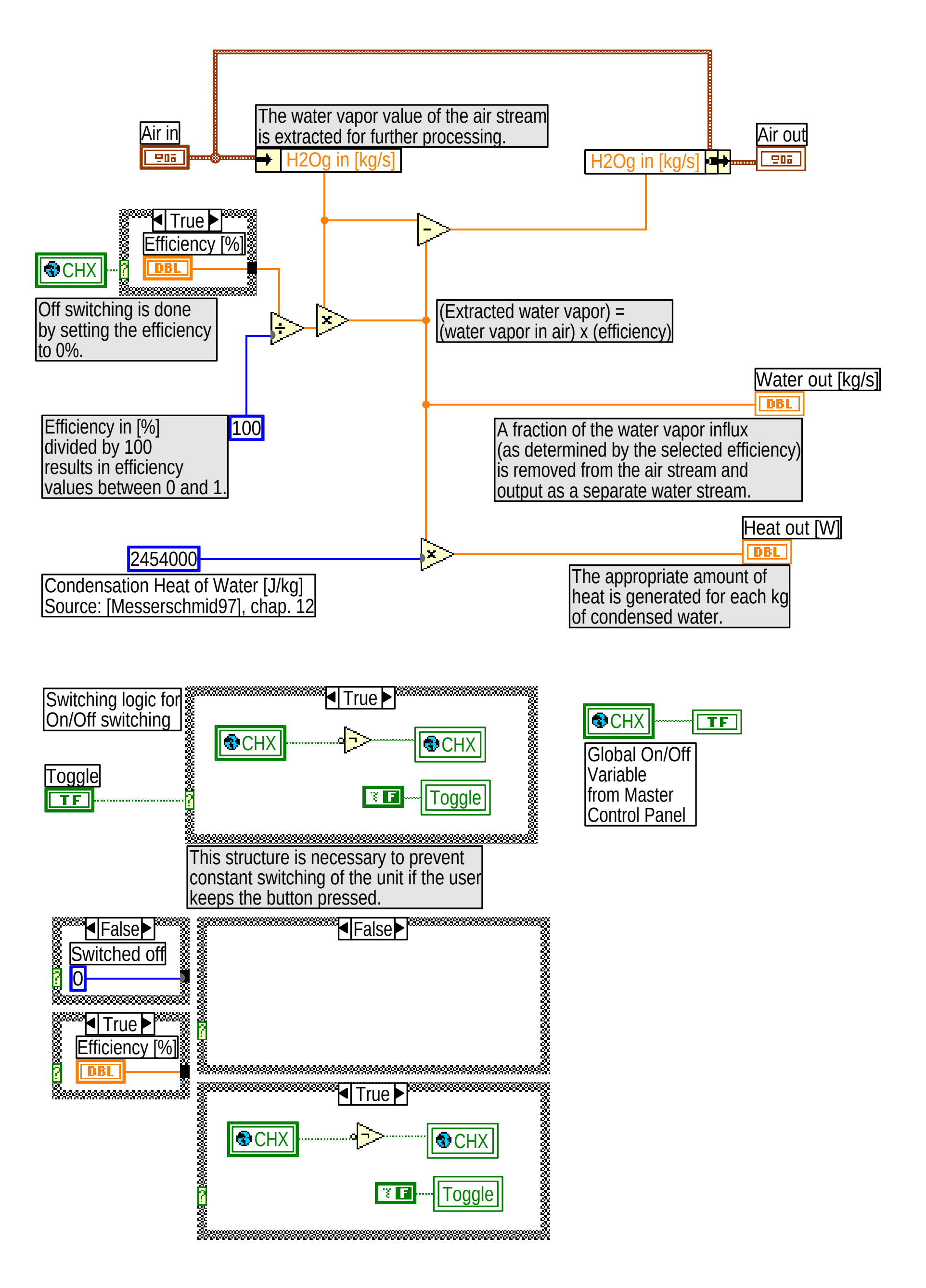


Abbildung 3.2.: Condensing Heat Exchanger Diagramm

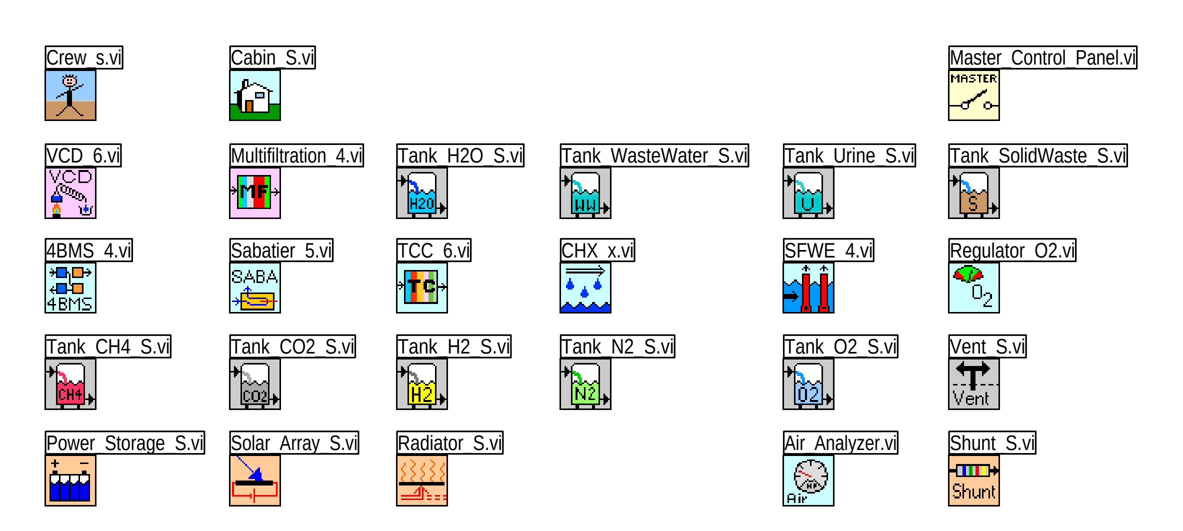


Abbildung 3.3.: VIs und Icons in der "ECLS Modules" Palette

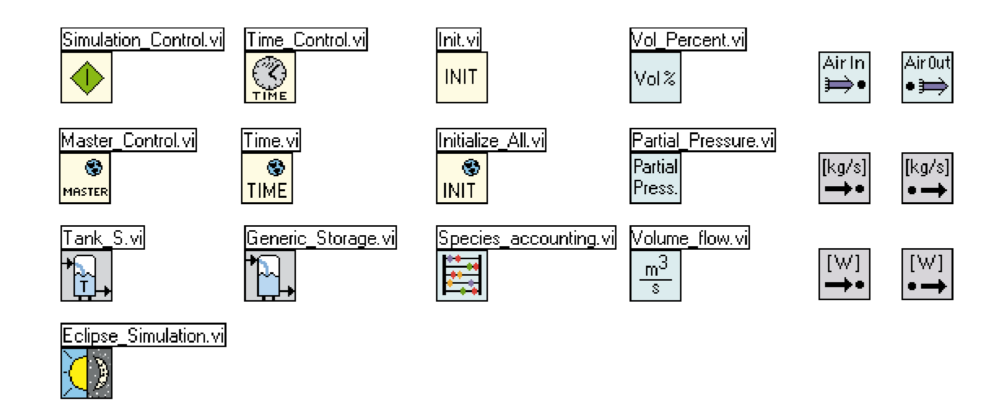


Abbildung 3.4.: VIs und Icons in der "ECLS Library" Palette

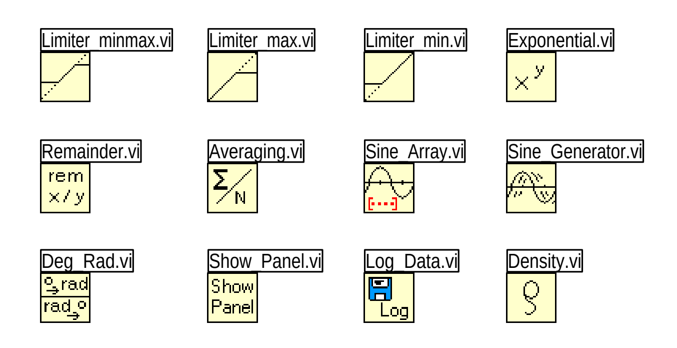


Abbildung 3.5.: VIs und Icons in der Auxiliary Library

3.2.2. Vorhandene Komponenten

Im Folgenden ist angegeben, welche ECLS- und EPS-Komponenten in der derzeitigen Version von MELISSA zur Verfügung stehen. Jeder Eintrag enthält Genaueres zur Art der Modellierung (soweit sie von den in Abschnitt 3.2.1 beschriebenen Grundsätzen abweicht), spezifiziert die Ein- und Ausgänge und benennt die jeweiligen VIs. Zu beachten ist, daß die implementierungsbedingten oder durch Besonderheiten der Simulationsumgebung notwendigen zusätzlichen Ein- und Ausgänge, wie sie in Abschnitt 3.2.4 angesprochen werden, hier nicht angegeben sind. Diese können der Dokumentation der jeweiligen MELISSA-VIs in Anhang C entnommen werden. Dort sind auch alle VIs der hier nicht berücksichtigten ECLS-Library- und Auxiliary-Library-Bibliotheken beschrieben.

3.2.2.1. Wasserkreislauf

Destillation: Eine Destillationseinheit (*Vapor Compressed Distillation*, VCD) dient zur Gewinnung von Brauchwasser aus Urin oder anderen verschmutzten Flüssigkeiten. Die modellierte Einheit ist für die Aufbereitung der gesamten Urin- und Schmutzwasserproduktion von sechs Personen ausgelegt. Falls sie nur zur Urinaufbereitung verwendet wird, kann sie ungefähr fünfzehn Personen versorgen. Eingänge sind Schmutzwasser und Energie, Ausgänge sind gereinigtes Wasser, feste Rückstände (*brine*) und Wärme. Diese Komponente ist als VCD_6.vi implementiert.

Multifiltrierung: Zur Rückgewinnung von Trinkwasser aus Schmutzwasser wird auch die Multifiltrierung (MF) eingesetzt. Das hier verwendete Gerät ist für vier Personen ausgelegt. Ein Wirkungsgrad von 99.9% wird aufgrund der geschlossenen Prozeßführung angenommen. Eingänge sind Schmutzwasser und Energie, Ausgänge Trinkwasser und Wärme. Diese Komponente ist als Multifiltration_4.vi implementiert.

Tanks: Für den Wasserkreislauf stehen Trinkwasser-, Schmutzwasser- und Urintanks sowie ein Lagerbehälter für feste Abfallstoffe (wie z. B. Fäkalien) zur Verfügung. Die Größe dieser Tanks kann frei gewählt werden. Eingang ist ein Massenstrom, Ausgänge sind ein Massenstrom und eine den Füllstand anzeigende Variable. Die Komponenten sind als Tank_H20.vi, Tank_WW.vi, Tank_U.vi, bzw. Tank_SW.vi implementiert.

3.2.2.2. Luftkreislauf

Analysator: Diese Komponente zeigt die Massenströme, die Volumenströme, die Volumenanteile und die Partialdrücke der Spezies eines Luftstroms (N_2, O_2, CO_2, Wasserdampf, Spurengase) an. Gleichzeitig werden auch die gemeinsamen Zustandsvariablen Druck und Temperatur angegeben. Das Gerät ist als Air_Analyzer.vi implementiert.

Elektrolyse: Bei der Wasserelektrolyse (*Static Feed Water Electrolysis*, SFWE) wird aus Wasser Sauerstoff und Wasserstoff hergestellt. Das modellierte Gerät ist zur Sauerstoffversorgung von vier Personen ausgelegt. Eingänge sind Wasser und Energie, Ausgänge Sauerstoff, Wasserstoff und Wärme. Diese Komponente ist als SFWE_4.vi implementiert.

Luftentfeuchter: Dieses Gerät (*Condensing Heat Exchanger*, CHX) entfernt einen einstellbaren Bruchteil des Wasserdampfes aus einem Luftstrom und gibt diesen als separaten Wasserstrom wieder aus. Die bei dem Prozeß freiwerdende Kondensationswärme wird im Modell berechnet. Der einzige Eingang ist der Luftstrom. Ausgänge sind der entfeuchtete Luftstrom, Wärme und Kondenswasser. Der gewünschte Wirkungsgrad kann auf dem Front Panel eingestellt werden. Diese Komponente ist als CHX_x.vi implementiert.

Sabatier: Ein Sabatier-Reaktor dient zur Umwandlung von CO_2 zu CH_4 und H_2O unter Verwendung von zusätzlichem H_2. Der modellierte Reaktor ist für fünf Personen ausgelegt. Eingänge sind Kohlendioxyd, Wasserstoff und Energie. Ausgänge sind Methan, Wasser und Wärme. Der Reaktor ist als Sabatier_5.vi implementiert.

Sauerstoff-Regulator: Dieses Gerät hält den Sauerstoff-Partialdruck in der Luft auf einem gewählten Wert, indem bei Bedarf zusätzlicher Sauerstoff zugefügt wird. Eingänge sind ein Luft- und ein Sauerstoffstrom, Ausgang ist ein Luftstrom. Dieses Gerät ist als Regulator_O2.vi implementiert.

Spurengas-Filter: Die in der Atemluft enthaltenen Spurengase (meist organische Verbindungen) werden von diesem Filter entfernt. Das modellierte Gerät ist für sechs Personen ausgelegt. Eingänge sind ein Luftstrom und Energie, Ausgang ist der gereinigte Luftstrom. Diese Komponente ist als TCC_6.vi implementiert.

Tanks: Für den Luftkreislauf stehen Tanks für Kohlendioxyd, Methan, Sauerstoff, Stickstoff sowie Wasserstoff zur Verfügung. Die Größe dieser Tanks kann frei gewählt werden. Eingang ist jeweils ein Massenstrom, Ausgänge sind ein Massenstrom und eine den Füllstand anzeigende Variable. Die Komponenten sind als Tank_CO2.vi, Tank_CH4.vi, Tank_O2.vi, Tank_N2.vi und Tank_H2.vi implementiert.

3.2.2.3. Energieversorgungssystem (EPS)

Solarzellen: Die Leistung der photovoltaischen Energieerzeugungsanlage hängt von den Faktoren Zellenfläche, Zellenmaterial (Gallium-Arsenid, Gallium-Phosphid oder Indium-Phosphid) und Zellenalter ab. Eingänge sind der solare Strahlungsfluß in $[W/m^2]$ sowie dessen Einfallswinkel. Ausgang ist die erzeugte Leistung. Diese Komponente ist als Solar_Array_s.vi implementiert.

Speichereinheit: Die Energiespeicherung kann wahlweise durch Akkumulatoren (Nickel-Cadmium oder Nickel-Wasserstoff) oder Schwungräder erfolgen. Im ersteren Fall

geht ein Wirkungsgradverlust in die Berechnung ein. Eingang ist die (z. B. von einem Solarzellen-Array) erzeugte Leistung, Ausgänge sind die abgegebene Leistung und eine Variable, die die im Modul gespeicherte Energie angibt. Die Speichereinheit ist als Power_Storage_s.vi implementiert.

Widerstand: Zur Umwandlung nicht benötigter elektrischer Energie in Wärme steht ein Widerstand einstellbarer Leistungsfähigkeit (*Shunt*) zur Verfügung. Der Shunt wird automatisch aktiviert, wenn die Energie, die im vorgeschalteten Energiespeicher vorhanden ist, einen vorgebbaren Grenzwert überschreitet. Eingang ist elektrische Leistung. Ausgänge sind Wärme sowie eine Variable, die die insgesamt umgewandelte Energie anzeigt. Diese Komponente ist als Shunt_s.vi implementiert.

3.2.2.4. Allgemeine Komponenten

Ablaßventil: Diese Komponente (*Vent*) dient zum Überbordgeben von überschüssigen Flüssigkeiten oder Gasen. Falls der Inhalt des vorgeschalteten Tanks über einen einstellbaren Grenzwert steigt, wird ein Teil des zugehörigen Spezies-Massenstroms mit vorgebbarer Rate über Bord geleitet. Eingang ist ein Massenstrom, Ausgang eine Variable, die die insgesamt über Bord gegebene Masse anzeigt. Das Ablaßventil ist als Vent_s.vi implementiert.

Besatzung: Die Anzahl und das Komfortniveau der Crew beeinflussen die Massen- und Wärmebilanz eines Lebenserhaltungssystems. In der verwendeten vereinfachten Modellierung sind die Verbrauchs- und Ausscheidungsraten abweichend vom wirklichen Verhalten unabhängig voneinander, d. h. mangelnde Zufuhr von z. B. Trinkwasser führt nicht zu einer Abnahme der Ausscheidungsrate von z. B. Urin, und so fort. Eingänge sind Luft, Trinkwasser und Nahrung. Ausgänge sind Schmutzwasser, Urin, Fäkalien und Wärme. Der Einfluß der Besatzung wird durch Verwendung des Crew_s.vi berücksichtigt.

Hauptschaltfeld: Dieses bietet eine Möglichkeit zum zentralisierten Ein- und Ausschalten aller Komponenten des Lebenserhaltungs- und Energieversorgungssystems. Es ist als Master_Control_Panel.vi implementiert.

Kabine: Die Größe des druckbeaufschlagten Volumens beeinflußt auch die Funktion des Lebenserhaltungssystems. Die Umwälzungsrate der Kabinenatmosphäre spielt ebenfalls eine Rolle. Die Modellierung erlaubt, dieses Volumen sowie die anfängliche Luftzusammensetzung vorzugeben. Ein- und Ausgang dieser Komponente ist ein Luftstrom. Die Kabine ist als Cabin_s.vi implementiert.

Radiator: Zur Wärmeabfuhr durch Abstrahlung werden Radiator-Elemente einstellbarer Größe verwendet. Aufgrund der vereinfachten Modellierung hängt der Wirkungsgrad des Prozesses im Gegensatz zu den wirklichen Verhältnissen nicht von

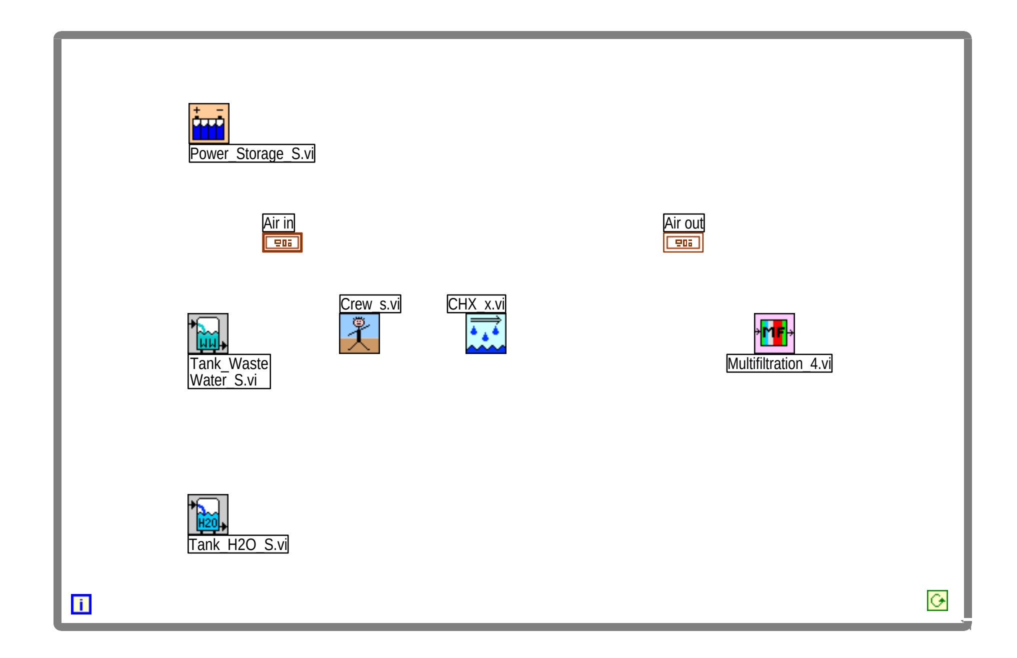


Abbildung 3.6.: Erster Schritt der Modellierung: Einfügen der nötigen Module

den thermodynamischen Zustandsgrößen des Kühlmediums oder den Bedingungen an der Radiatoroberfläche ab, sondern läßt sich vorgeben. Eingang ist ein Wärmestrom, Ausgänge sind der verbleibende Wärmestrom. Diese Komponente ist als Radiator_s.vi implementiert.

3.2.3. Modellierung und Simulation

Die folgende kurze Beschreibung der Vorgehensweise bei der Modellierung von Systemen mit Melissa dient zum besseren Verständnis der gewählten Lösung. Eine ausführliche Darstellung einschließlich einer Schritt für Schritt bearbeiteten Beispielaufgabe findet sich im Melissa-Benutzerhandbuch in Anhang E.

Das graphische Erstellen eines Systemmodells erfolgt in mehreren Schritten. Zuerst werden im Diagramm-Fenster die im zu modellierenden Lebenserhaltungssystem verwendeten Baugruppen (z. B. Wärmetauscher, Wasserfilter, ...) eingefügt. Abbildung 3.6 zeigt die Momentaufnahme einer entstehenden Beispiel-Simulation nach diesem Schritt. Es handelt sich hier um die in Anhang C dokumentierte "Example.VI"-Simulation. Erkennbar sind Module für die Energie- und Luftversorgung, Tanks für Trink- und Schmutzwasser, die Crew, ein Wärmetauscher (*CHX*) und ein Wasserfilter. Diese und alle anderen von Melissa zur Verfügung gestellten Module wurden oben bereits beschrieben.

Diagramm

Die Ein- und Ausgänge dieser Komponenten werden dann zur Definition der Stoff- und Energieströme mittels "Kabeln" (*wires*) verbunden, worauf in Abschnitt 3.2.4 noch ge-

Wires

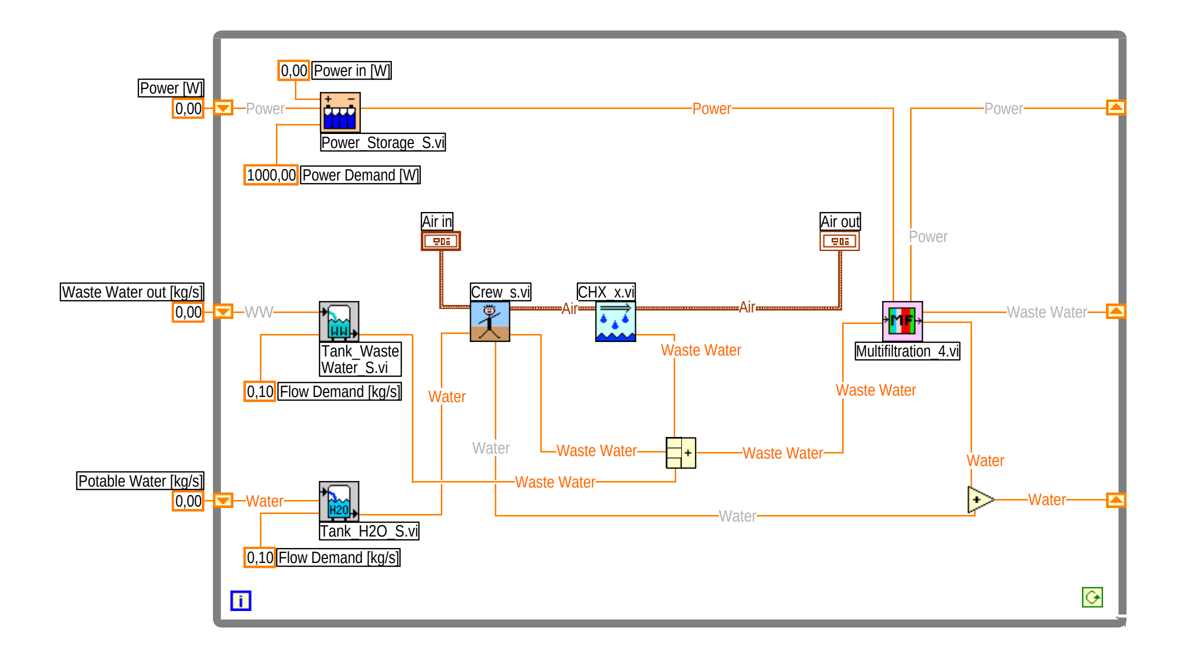


Abbildung 3.7.: Nächster Schritt: Anschließen der Komponenten

nauer eingegangen wird. Abbildung 3.7 zeigt das Aussehen des Simulationsmodells im Diagramm nach Beenden dieses Schrittes.

Plausibilitäts-kontrolle

Durch entsprechende Festlegungen in den Komponenten-VIs ist sichergestellt, daß der Benutzer beim Verbinden der Module die notwendigen Anschlüsse vornimmt (*required connections*, siehe Benutzerhandbuch im Anhang). Zusätzlich kann im Hilfefenster angezeigt werden, an welchen Ausgang eines VIs ein Draht angeschlossen ist. Auf diese Weise kann die korrekte Verkabelung überprüft werden.

Front Panel

Nach Erstellen des Diagramms werden die zur Anzeige von Simulationswerten benötigten Schnittstellen auf dem Front Panel eingefügt und angeschlossen. Hier können die von LabVIEW zur Verfügung gestellten Instrumentenbibliotheken benutzt werden, die – wie in Abbildung 3.8 ersichtlich – vertraute Formen wie Digitalanzeigen oder Analoginstrumente enthalten.

Abschließend können noch Hilfsmodule, die im Benutzerhandbuch detailliert beschrieben sind, zur Vereinfachung des Simulationsablaufs eingefügt werden.

Bedienung

Sobald das MELISSA-Modell eines Systems fertiggestellt ist, kann mit Simulationsläufen begonnen werden. Dazu werden zunächst im Systemdiagramm und auf den Front Panels der beteiligten Module die gewünschten Anfangsbedingungen eingestellt, z. B. anfängliche Tankinhalte, Crewanzahl oder Luftzusammensetzung. Nach Starten der Simulation kann der Benutzer auf den Front Panels der Simulations-VI und ihrer Komponenten das Fortschreiten der Rechnungen verfolgen und gegebenenfalls steuernd eingreifen. Gewonnene Zwischen- und Endergebnisse können für jeden Lauf in einer Datei

Auswertung

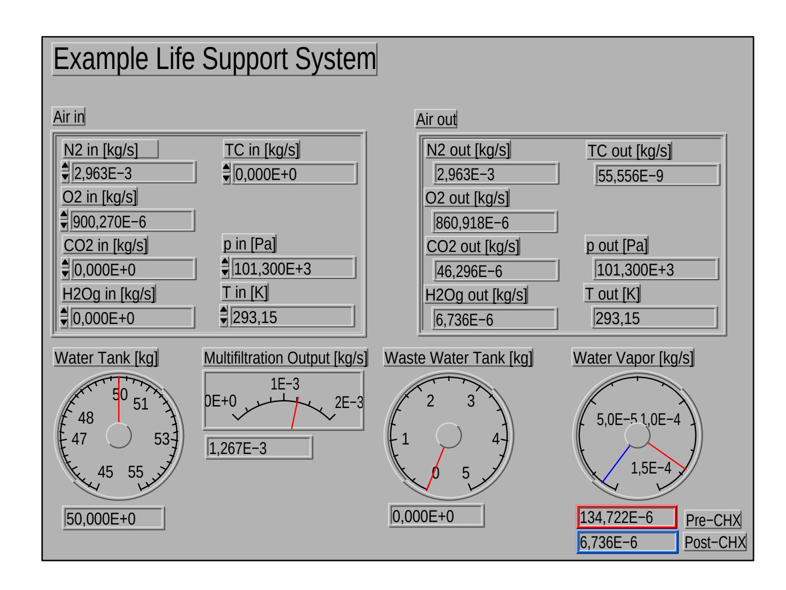


Abbildung 3.8.: Front Panel mit Digital- und Analoganzeigen

gespeichert und mit herkömmlichen Datenverarbeitungsprogrammen wie MS Excel oder gnuplot zusätzlich ausgewertet und visualisiert werden.

3.2.4. Besonderheiten der Simulationsumgebung

Wie jedes Computerprogramm hat LabVIEW und damit auch MELISSA Besonderheiten, die nicht auf Anhieb intuitiv zu erfassen sind. Obwohl die verwendete graphische Oberfläche mit ihrer “Schaltbild”-Analogie dem Benutzer das Erstellen und Ausführen von Simulationen erleichert, sind einige Abweichungen von dieser Analogie erklärungsbedürftig. Die folgenden Ausführungen dienen damit auch dem besseren Verständnis der Möglichkeiten und Grenzen von MELISSA.

3.2.4.1. Unidirektionaler Datenfluß

Die Mehrzahl dieser Abweichungen wird von der Tatsache verursacht, daß die “Kabel”, die von LabVIEW zur Verfügung gestellt werden, im Gegensatz zu echten Leitungsverbindungen nur einen unidirektionalen Datenfluß ermöglichen. Die LabVIEW-Kabel definieren *eine* Quelle und eine oder mehrere dazugehörige Senken für bestimmte Informationen, z. B. Massenströme. Entlang eines solchen Kabels kann sich Information also nur in eine Richtung fortpflanzen. Dieser Unterschied hat weitreichende Folgen, vor allem im Zusammenhang mit den zu simulierenden Masse- und Energieströmen.

Beispiel Tank

Zur Verdeutlichung diene folgendes Beispiel: Ein Tank sei über eine Leitung mit einem Verbraucher verbunden, der von diesem Tank mit einem Massenstrom versorgt wird



Abbildung 3.9.: Tank und Verbraucher

Abbildung 3.10.: Geregelter Massenfluß

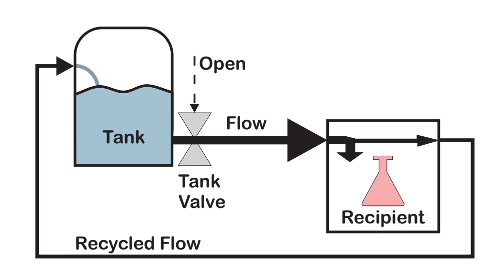


Abbildung 3.11.: "Recycling" des Massenflusses

(Abbildung 3.9). Sowohl am Tankausgang wie auch am Eingang des Verbraucher-Moduls werde der Massenstrom in der Leitung durch Ventile reguliert. Offensichtlich hängt der Massenstrom, der den Tank verläßt, nicht nur vom Tankventil, sondern genauso auch vom stromabwärts des Tanks angeordneten Ventil am Verbraucher-Modul ab. Dieses Verhalten beruht auf der Ausbreitung von Information in der Leitung entgegen der Strömungsrichtung, wie sie (zumindest bei Unterschallströmungen) selbstverständlich ist.

Falls nun aber die eigentliche Leitung diesen Informationstransport nicht bewerkstelligen kann, muß das Anpassen des den Tank verlassenden Massenstroms an die Bedürfnisse des Verbrauchers auf andere Weise geschehen. Beispielsweise kann der Verbraucher über eine Steuerleitung die gewünschte Einstellung des Tankventils vornehmen (Abbildung 3.10, *geregelter* Massenfluß).

Recyclingstrom

Eine andere Möglichkeit besteht darin, bei mehr als nötig geöffnetem Tankventil den vom Verbraucher nicht benötigten Anteil des Massenstroms wieder in den Tank zurückzuleiten (Abbildung 3.11, *Recycling*). Bei MELISSA wird letzteres Konzept verwendet, weil dies die Problematik von Reglern in einer zeitdiskreten Umgebung vermeidet. Daher besitzen die meisten MELISSA-Module – zusätzlich zu den in Abschnitt 3.2.2 angegebenen – Ausgänge, über die der nicht verbrauchte oder verarbeitete Rest der eingehenden Massen- oder Energieströme wieder zurückgegeben wird. Außerdem muß an jedem Tank- oder Speicherelement ein Sollwert für den abfließenden Massen- oder Energiestrom angegeben werden (*Flow Demand Value*).

3.2.4.2. Serielles Verbinden von Modulen

Von der Problematik des Informationsflusses ist auch ein weiteres erklärungsbedürftiges Modellierungsprinzip abgeleitet: Mehrere Geräte, die vom gleichen Tank versorgt werden, müssen *seriell* an diesen angeschlossen werden. Auch hier soll ein Beispiel zur Verdeutlichung dienen: Normalerweise werden mehrere Geräte, die aus einer Leitung versorgt werden, mittels eines T-Stücks oder eines ähnlichen Elementes parallel angeschlossen (Abbildung 3.12). Dies setzt jedoch wieder einen stromaufwärts gerichteten Informationsfluß voraus, damit die gemeinsam benötigte Menge aus dem Tank strömen kann und sich der Massenstrom dann am T-Stück entsprechend dem Bedarf aufteilt.

In MELISSA läßt sich dies wegen des von LabVIEW vorgegebenen unidirektionalen Informationsflusses wiederum nicht übernehmen, so daß nur eine serielle Schaltung wie in Abbildung 3.13 in Frage kommt. Hier erhält der zweite Verbraucher den vom ersten übriggelassenen Massestrom, entnimmt daraus den von ihm benötigten Anteil und leitet den Rest wieder gemäß dem oben eingeführten Recycling-Konzept an den Tank zurück.

3.2.4.3. Rückführen der überschüssigen Massenströme

Beim Rückführen von überschüssigen Massenströmen muß berücksichtigt werden, daß kein LabVIEW-Diagrammelement – ob einfacher Operator oder komplexe Sub-VI – Eingänge haben darf, die ihren Ursprung, vom Datenfluß her betrachtet, "stromabwärts" des Elementes haben, da dies die Schaffung eines arithmetisch nicht auflösbaren Zirkelbezugs bedeuten würde. Diese Zirkelbezüge führen selbst in spezialisierten verfahrenstechnischen Simulatoren (wie z. B. dem in Kapitel 2 auf Seite 9 vorgestellten AspenPlus) zu Problemen und sollten generell vermieden werden [25]. Das hat zur Folge, daß die "Recycling"-Leitung auf dem Diagramm nicht einfach wieder zurück zum Tank gelegt werden kann.

Deshalb wird im betrachteten Fall der zurückzuführende Massenstrom zwischengespeichert und dann erst zu Anfang des nachfolgenden Simulationsschrittes in den entsprechenden Tank weitergeleitet. Für dieses Verfahren stellt LabVIEW im Zusammenhang mit der zur wiederholten Ausführung der Simulationsschritte verwendeten "While"-Schleife sogenannte *Shift Register* zur Verfügung, die in Abbildung 3.14 zu sehen sind.

Shift Register

3.2.5. Benutzerhandbuch

Das im Rahmen dieser Arbeit erstellte MELISSA-Benutzerhandbuch (Anhang E ab Seite 247) enthält weitere, detaillierte Informationen zu Bedienung und Besonderheiten von MELISSA. Es bietet auch eine Schritt-für-Schritt-Anleitung für Benutzer, die eigene Simulationen erstellen wollen, sowie ausführliche Design-Leitlinien für die Modifikation vorhandener oder den Entwurf neuer MELISSA-Module.

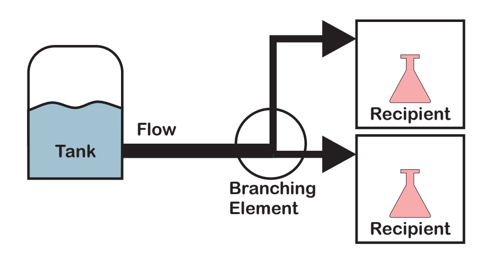


Abbildung 3.12.: Paralleler Anschluß zweier Verbraucher an einen Tank

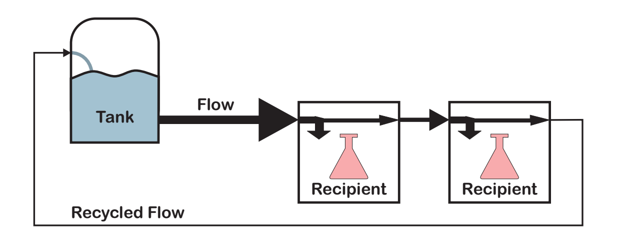


Abbildung 3.13.: Serieller Anschluß zweier Verbraucher an einen Tank

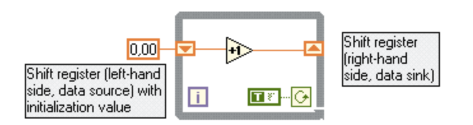


Abbildung 3.14.: LabView "While"-Schleife mit Shift Register: Der nach Ausführung der Schleife im Draht vorhandene Wert wird zu Beginn des nächsten Schleifendurchlaufs wieder "stromauf" eingespeist und auf diese Weise mit jedem Schleifendurchlauf um eins erhöht.

4. Beispielanwendung: Internationale Raumstation

Die Internationale Raumstation (ISS) wird bis weit über das Jahr 2000 hinaus das wichtigste Anwendungsfeld für Technologien der bemannten Raumfahrt sein. Die Analyse ihrer System- und Subsystemeigenschaften stellt eine typische Aufgabe für den Space Station Design Workshop dar und bietet sich somit auch zur Überprüfung der Fähigkeiten der entwickelten Entwurfs- und Simulationsumgebung an. Deshalb wird hier die Untersuchung des ISS-Lebenserhaltungssystems als erstes Anwendungsbeispiel für MELISSA beschrieben. Aus Gründen der Datenverfügbarkeit beschränkt sich die Modellierung auf den US-Teil der Station (United States Orbital Segment, USOS).

ISS

Die zugrundeliegenden Daten stehen größtenteils elektronisch zur Verfügung [9], können aber auch konventionellen Publikationen wie dem Werk [32] entnommen werden. Hifreich ist in diesem Zusammenhang, daß sich die meisten implementierten MELISSA-Module bereits an ISS-Hardware anlehnen.

Nach der Modellierung, die im nächsten Abschnitt beschrieben ist, werden die Ergebnisse von Untersuchungen zum Betriebsverhalten unter Normalbedingungen sowie bei Ausfall einzelner Teilsysteme präsentiert. Solche Simulationen sind zur Qualifikation eines ECLSS-Konzeptes notwendig und dienen zum Nachweis spezifikationsgemäßen Subsystemverhaltens. Die Ausfalluntersuchungen sind nach den jüngsten Schwierigkeiten an Bord der russischen Raumstation Mir (Ausfall des Sauerstoffgenerators [18] und des CO_2-Filters [19]) von zusätzlichem Interesse. Die Ergebnisse der Simulationsläufe werden in den Abschnitten 4.2 und 4.3 dokumentiert und in Abschnitt 4.4 diskutiert.

4.1. Aufbau des Simulationsmodells

Abbildung 4.1 zeigt ein Übersichtsbild des zu modellierenden ECLSS, welches auf den in den oben angegebenen Quellen enthaltenen Informationen basiert.

Die Bestandteile des Lebenserhaltungssystems sind in der gezeigten Endausbaustufe auf mehrere Module der Raumstation verteilt. Die Sauerstoff- und Stickstofftanks sind aus Platz- und Sicherheitsgründen außen an der Luftschleuse angeordnet. Die Hauptmodule für Labor (*Lab*) und Wohnen (*Hab*) verfügen jeweils über einen eigenen Luftkreislauf, der jedoch über das freie Kabinenvolumen mit den anderen Modulen in Verbindung

Luftkreislauf

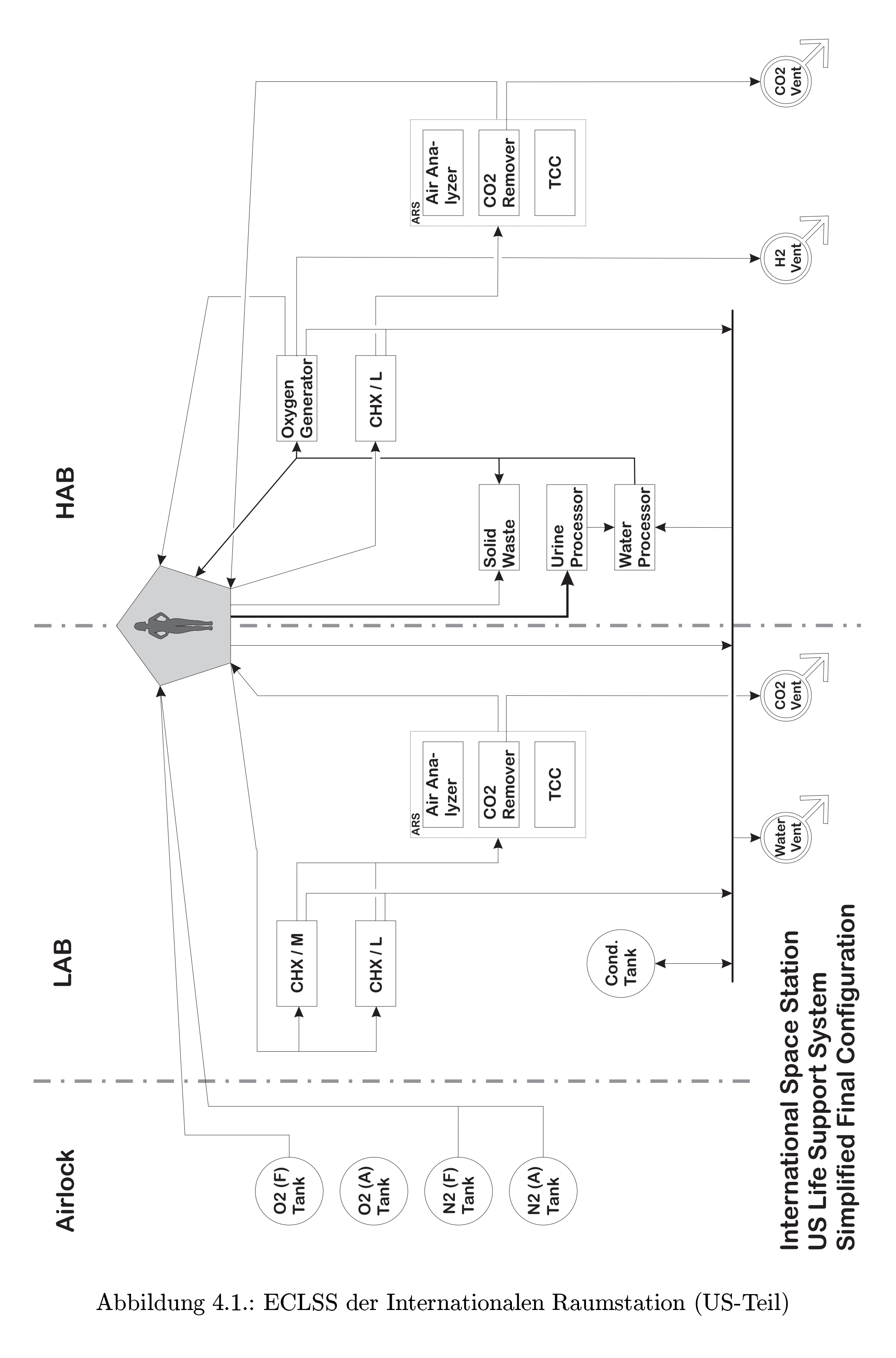


Abbildung 4.1.: ECLSS der Internationalen Raumstation (US-Teil)

steht. In diesem Kabinenkreislauf werden die Luftfeuchtigkeit, der CO_2-Gehalt und der Spurengasanteil geregelt. Im Wohnmodul befindet sich auch ein Sauerstoffgenerator, der nach dem Elektrolyseprinzip (SFWE) arbeitet.

Wasserkreislauf

Ein Rack des Labormoduls bietet Platz für einen Kondenswasserbehälter, der leicht verschmutztes Wasser wie z. B. aus der Luftentfeuchtung bis zur Weiterverarbeitung im Wasserfilter (*Multifiltration*) zwischenlagern kann. Dieses befindet sich ebenso wie ein Urin-Destillator (*Vapor Compressed Distillation*, VCD) und eine Toilette mit angeschlossenem Fäkalienbehälter im Wohnmodul. Der Trinkwasservorrat kann gegebenenfalls durch Übernahme von Wasser, das in den Brennstoffzellen des Space Shuttle produziert wird, ergänzt werden.

Aufgrund des konservativ ausgelegten Systems ist nur der Wasserkreislauf als weitgehend geschlossen anzusehen. Das aus der Luft entfernte Kohlendioxyd wird genauso über Bord gegeben wie der bei der Elektrolyse anfallende Wasserstoff. Auch der Schmutzwassertank kann auf diese Weise entleert werden, falls er die Grenze seiner Aufnahmekapazität erreicht.

Umsetzung

Das MELISSA-Modell des Lebenserhaltungssystems, zu sehen in Abbildung 4.2, ist entsprechend den Leitlinien aus Kapitel 2 erstellt. Die wesentlichen Unterschiede ergeben sich aus der gleichzeitigen Modellierung eines Energieversorgungssystems sowie aus der separaten Beschreibung diverser Tanks, die im Original-ECLSS in die entsprechenden Verarbeitungsgeräte integriert sind (Trinkwasser, Urin, Fäkalien). Außerdem sind einige für die Simulation notwendige Hilfsmodule auf dem Diagramm vorhanden (z. B. das Master Control Panel, Abbildung 4.11). Die Stickstofftanks sind nicht erfaßt, da die Simulation in ihrer derzeitigen Version keine Leckagen berücksichtigt und so mangels Stickstoffverbrauchern immer eine konstante Stickstoffmenge vorhanden ist.

Verfolgte Größen

Das Front Panel der ISS-Simulation (Abbildung 4.3) ist mit Anzeigen für die Füllstände der Tanks und den Zustand des Energiesystems versehen. Mehrfachinstrumente fassen jeweils die Tankinhalte der Schmutztanks (Schmutzwasser, Urin, Fäkalien) und der Vorratstanks (Trinkwasser, Sauerstoff) zusammen. Neben diesen Werten wird bei den Untersuchungen auch die Luftzusammensetzung betrachtet, die vom Air-Analyzer-Moduls (Abbildung 4.4) angezeigt wird.

Anfangswerte

Die Instrumente in Abbildung 4.3 und Abbildung 4.4 zeigen die Initialisierungswerte der Komponenten an. Diese und weitere Daten sind nachfolgend für alle verwendeten Module aufgeführt, um die Reproduzierbarkeit der Untersuchungen zu gewährleisten. Die Aufzählung umfaßt nur solche Parameter, die an den Front Panels der einzelnen Geräten vom Benutzer eingestellt werden können. Generelle Angaben über die Modellierung der Geräte und zugrundeliegende Annahmen sind den Dokumentationsausdrucken der MELISSA-Module in Anhang C ab S. 91 zu entnehmen. Ein Großteil der Werte ist direkt der Literatur entnommen, fehlende Angaben bedingen jedoch die Notwendigkeit von einigen Schätzungen. Die Aufzählung folgt der Anordnung der Geräte in Abbildung 4.1.

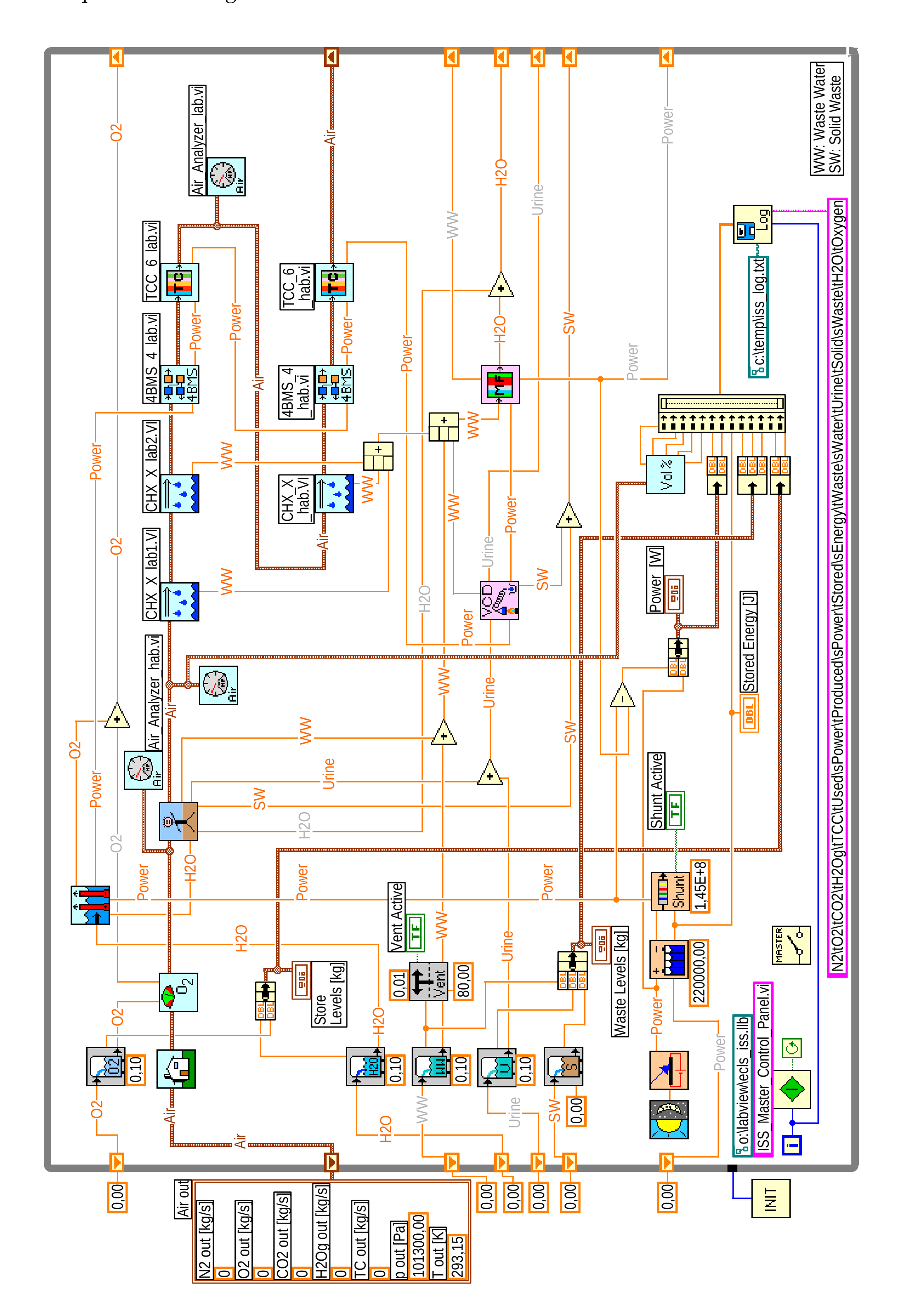


Abbildung 4.2.: MELISSA-Modell des ISS-Lebenserhaltungssystems (US-Teil)

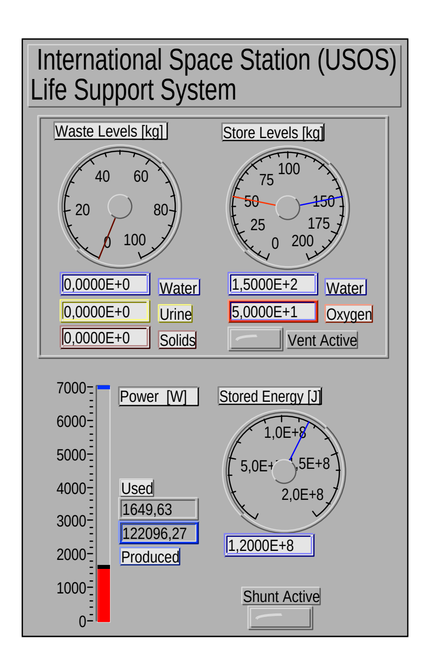


Abbildung 4.3.: Front Panel des MELISSA-Modells (mit Anfangswerten)

Air Analyzer

Air in

N2 in [kg/s] 9,875E−3
TC in [kg/s] 55,556E−9
O2 in [kg/s] 2,962E−3
CO2 in [kg/s] 46,296E−6
p in [Pa] 101,300E+3
H2Og in [kg/s] 134,722E−6
T in [K] 293,15

Total mass flow [kg/s] 13,018E−3
Total volume flow [m3/s] 10,086E−3

Volume flows		Volumetric fractions		Partial pressures	
N2 [m3/s]	7,900E−3	N2 [% Vol.]	78,329E+0	ppN2 [Pa]	79,347E+3
O2 [m3/s]	2,072E−3	O2 [% Vol.]	20,548E+0	ppO2 [Pa]	20,816E+3
CO2 [m3/s]	23,417E−6	CO2 [% Vol.]	232,184E−3	ppCO2 [Pa]	235,203E+0
H2Og [m3/s]	89,815E−6	H2Og [% Vol.]	890,514E−3	ppH2Og [Pa]	902,091E+0
TC [m3/s]	27,778E−9	TC [% Vol.]	275,417E−6	ppTC [Pa]	278,997E−3

Abbildung 4.4.: Anfängliche Luftzusammensetzung nach dem Crew-Modul

Crew: 4 Personen, mittelhohes Komfortniveau (*nice*)

Kabine: Volumen 1000 m^3, anfängliche Luftzusammensetzung 79% Stickstoff, 21% Sauerstoff, Umwälzungsrate 0,1%/s

Sauerstofftank: Kapazität 100 kg, anfänglicher Inhalt 50 kg

Sauerstoff-Generator: SFWE_4-Modul, Sauerstoffregulator auf optimalen O_2-Partialdruck von 21 kPa (vor Crew) eingestellt

Luftentfeuchter: CHX_x-Modul, Wirkungsgrad 95%

Kohlendioxyd-Entfernung: 4-Bett-Molekularsiebe (4-BMS_4), Wirkungsgrad 66%

Spurengasfilter: TCC_6-Modul, Wirkungsgrad 99%

Fäkalien: Tank-Kapazität 100 kg, anfänglicher Inhalt 0 kg

Urinprozessor: VCD_6-Modul, Tank-Kapazität 100 kg, anfänglicher Inhalt 0 kg

Schmutzwassertank: Kapazität 100 kg, anfänglicher Inhalt 0 kg

Wasserprozessor: Multifiltration_4-Modul, Tank-Kapazität (Trinkwasser) 200 kg, anfänglicher Inhalt 150 kg

Wasser-Ablaß: Vent_s-Modul, maximaler Durchfluß 1 kg/s, automatische Aktivierung bei 80 kg Schmutzwassertank-Inhalt

Kohlendioxyd-Ablaß: direktes Überbordgeben am 4-BMS

Wasserstoff-Ablaß: direktes Überbordgeben am Elektrolysegerät

Das zusätzlich modellierte Energiesystem ist ohne Bezug zur ISS ausgelegt; es dient nur dazu, die ECLSS-Komponenten während der Simulation mit korrekter Eingangsleistung zu versorgen. Die Bahnsimulation beruht jedoch auf der ISS-Spezifikation. Die relevanten Daten sind:

Energiespeicher: NiCd-Akkumulatoren, Kapazität $150 \cdot 10^6$ J, anfänglich gespeicherte Energie $120 \cdot 10^6$ J

Solarzellen: Silizium, Fläche 1000 m^2, Alter 3 Jahre, Bestrahlungswinkel 0°(d. h. α- und β-Nachführung der Arrays)

Bahnsimulation: Bahnhöhe 400 km, Inklination 51.6°, Eklipse-Berechnungen für den ungünstigsten Fall (*worst case*, entsprechend einem Eklipse-Faktor von 37,96%)

Shunt: maximale Energieabgabe $2 \cdot 10^5$ W, automatisches Einschalten bei $140 \cdot 10^6$ J gespeicherter Energie

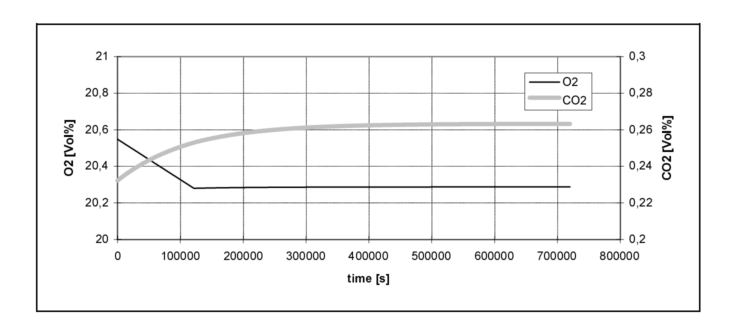


Abbildung 4.5.: Anfänglicher Verlauf der Luftzusammensetzung

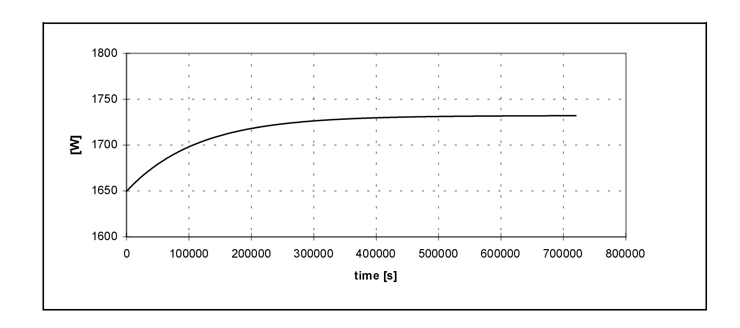


Abbildung 4.6.: Anfänglicher Verlauf des Energieverbrauchs

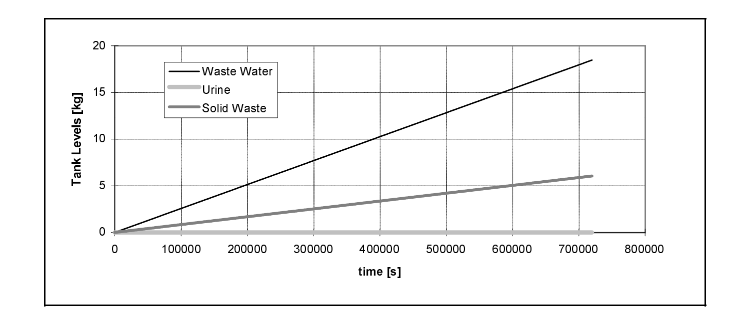


Abbildung 4.7.: Anfänglicher Verlauf der Schmutztank-Inhalte

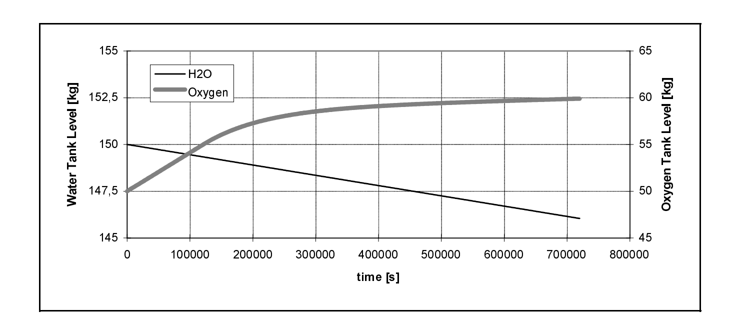


Abbildung 4.8.: Anfänglicher Verlauf der Vorratstank-Inhalte

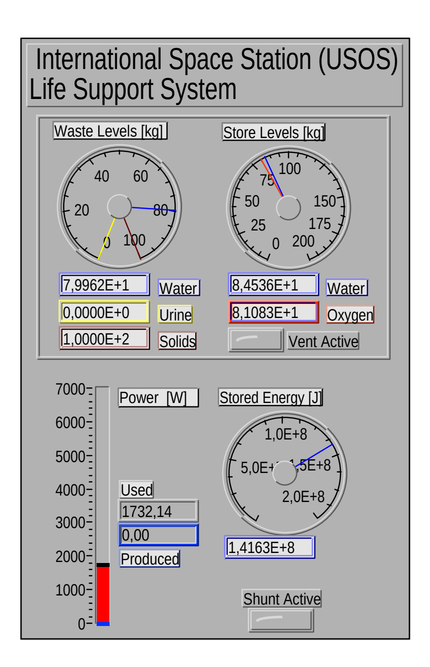


Abbildung 4.9.: Front Panel mit Tankinhalten nach 138 Tagen

4.2. Langzeitsimulation

Überlauf

Die oben beschriebenen Einstellungen sind Grundlage der durchgeführten Langzeitsimulation des ECLSS der Internationalen Raumstation. Alle vorhandenen Module bleiben dabei angeschaltet und arbeiten mit normaler Kapazität. Die anfänglichen Verläufe der Luftzusammensetzung, des Energieverbrauchs sowie der Tankinhalte sind in den Abbildungen 4.5, 4.6, 4.7 und 4.8 dargestellt. Nach ca. 138 Tagen kommt es durch Überlauf des Fäkalientanks, der seine maximale Kapazität von 100 kg erreicht hat, zum Abbruch der Simulation. Durch regelmäßiges Leeren des Tanks, wie in den Betriebsplänen der ISS vorgesehen, läßt sich dieser Abbruch natürlich vermeiden und die Simulation somit weiter fortsetzen. Die Tankinhalte zu diesem Zeitpunkt sind auf dem Front Panel in Abbildung 4.9 zu erkennen, während Abbildung 4.10 die entsprechende Luftzusammensetzung zeigt.

4.3. Ausfalluntersuchungen

Neben der Analyse des Langzeitverhaltens des Lebenserhaltungssystems unter Normalbedingungen gehört auch die Untersuchung der Auswirkungen von Geräteausfällen zum Umfang der Qualifikation eines ECLSS. Ein Geräteausfall kann im Rahmen der Melissa-Simulation durch Abschalten des entsprechenden Moduls mittels des in Abbildung 4.11 gezeigten Hauptschaltfeldes herbeigeführt werden.

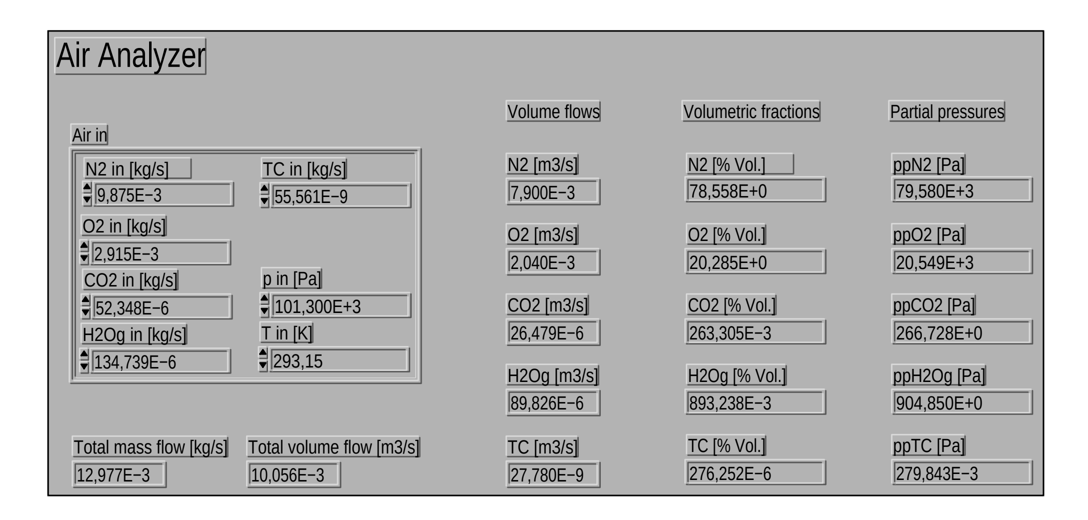


Abbildung 4.10.: Luftzusammensetzung nach 138 Tagen

Die jeweils wichtigsten Geräte des Wasser- bzw. Luftkreislaufes sind das Wasserfilter bzw. die Molekularsiebe zur Kohlendioxydentfernung. Ferner ist im ungünstigsten Fall ein Totalausfall des ECLSS möglich. Aus diesem Grunde sind hier die Ergebnisse von entsprechenden Simulationsläufen wiedergegeben.

Die zugrundeliegenden Simulationsrechnungen gehen von Anfangswerten für die Füllstände der Tanks und für die Luftzusammensetzung aus, wie sie von einem normal arbeitenden System nach der Anlaufphase (ca. nach 8 Tagen) erreicht werden. Dies kommt den Bedingungen bei einem Ausfall während des laufenden Betriebes am nächsten. Die verwendeten Werte sind in den Abbildungen des vorherigen Abschnittes dokumentiert. Gegenüber der Aufstellung auf Seite 36 ergeben sich dadurch folgende Änderungen:

Aus laufendem Betrieb

Kabine: anfängliche Luftzusammensetzung mit 20,287% Sauerstoff

Sauerstofftank: anfänglicher Tankinhalt 59,905 kg

Fäkalien: anfänglicher Tankinhalt 6,0526 kg

Schmutzwassertank: anfänglicher Tankinhalt 18,487 kg

Wasserprozessor: anfänglicher Tankinhalt 146,04 kg

4.3.1. Ausfall des Wasserfilters

Bei einer Unterbrechung des Wasserkreislaufes durch Ausfall von dessen wichtigster Komponente, des Wasserfilters (MF), kommt es nach 109550 s (ca. 30,5 h) zum Abbruch der Simulation aufgrund des leeren Trinkwassertanks. Abbildung 4.12 zeigt den zeitlichen Verlauf der Trink- und Schmutzwassertank-Inhalte.

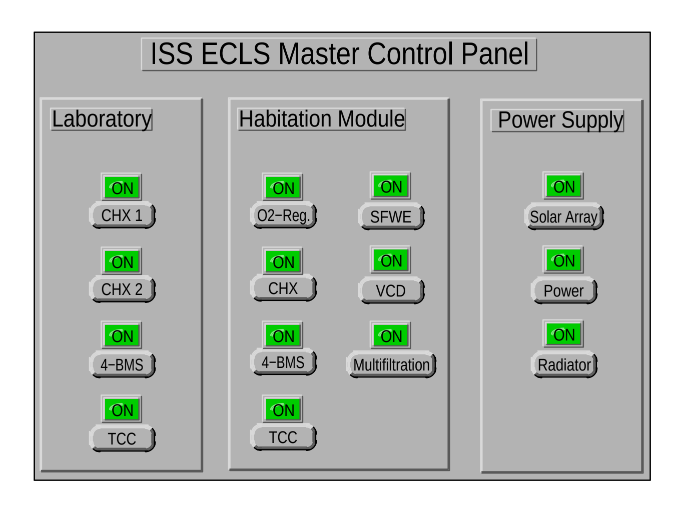


Abbildung 4.11.: Hauptschaltfeld des Raumstations-ECLSS

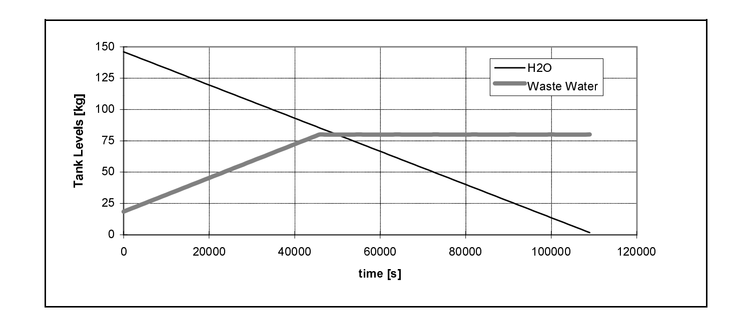


Abbildung 4.12.: Wassertanks bei ausgefallenem Filter

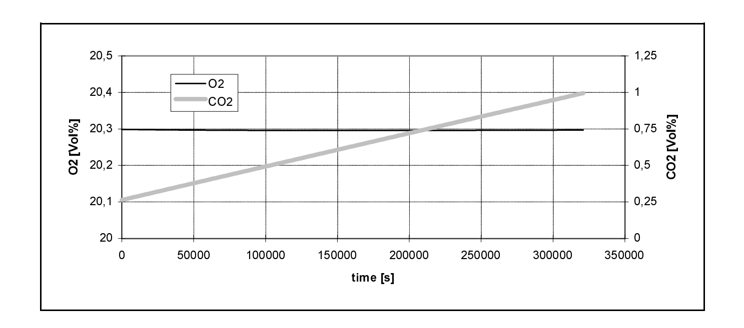


Abbildung 4.13.: Verlauf der Luftzusammensetzung bei ausgefallenen Molekularsieben

4.3.2. Ausfall der Molekularsiebe

Ein Ausfall der Molekularsiebe[1], wie er sich beispielsweise während der Erstellung dieses Berichtes auf der russischen Raumstation Mir ereignete [19], führt bei dem modellierten System nach 321680 s (ca. 89 h) zum Überschreiten des Grenzwertes für den CO_2-Partialdruck in der Luft. Dieser ist für die Internationale Raumstation mit 1010 Pa ([34], Anhang B.1) spezifiziert. Die Auswirkungen von Notmaßnahmen seitens der Crew, wie der Einsatz von Lithiumhydroxyd zur Kohlendioxyd-Bindung, sind dabei jedoch nicht in Betracht gezogen. Abbildung 4.13 zeigt den zeitlichen Verlauf der Luftzusammensetzung.

4.3.3. Totalausfall des ECLS-Subsystems

Der völlige Ausfall aller ECLS-Komponenten kann z. B. beim Ausfall der Energieversorgung eintreten. Ein solcher Zwischenfall hat sich ebenfalls bereits auf der Raumstation Mir ereignet. Er wird hier als *worst case* in Betracht gezogen. Deshalb ist für diese Untersuchung auch die Crew-Anzahl von 4 auf 6 erhöht, um ein notfallbedingtes Mitversorgen der Bewohner des russischen Teils der Internationalen Raumstation zu berücksichtigen; gleichzeitig ist das Komfort-Niveau auf den niedrigsten Wert (*spartan*) gesetzt.

In der Simulation unterschreitet der Sauerstoffpartialdruck der Kabinenluft den Grenzwert von 19,5 kPa (siehe wiederum [34], Anhang B.1) nach 234980 s (ca. 65 h). Auch der CO_2-Partialdruck bewegt sich zu diesem Zeitpunkt schon auf seinen Grenzwert zu. Die Verläufe der Luftzusammensetzung und der Tankinhalte sind in den Abbildungen 4.14, 4.15 und 4.16 dargestellt.

4.4. Diskussion der Ergebnisse

Die Untersuchungen zeigen, daß das Modellieren von Lebenserhaltungssystemen und die Durchführung von Simulationsrechnungen mit diesen Modellen sowie deren anschließende Auswertung mit geringem Aufwand möglich sind. Die notwendige Rechenzeit hält sich ebenfalls in Grenzen: Die Langzeitsimulation lief über Nacht; die Ausfallsimulationen benötigten auf dem verwendeten Rechner (Pentium-PC, 60 MHz, 32 MB RAM, Windows 3.1) jeweils weniger als eine Stunde.

Die Ergebnisse weisen keine Abweichungen von den zu erwartenden Verläufen auf, so daß von korrekter Modellierung und Simulation ausgegangen werden kann. Mangels vorliegender Betriebs- oder Testdaten des tatsächlichen ISS-Lebenserhaltungssystems kann jedoch noch keine weitergehende Verifikation durchgeführt werden.

[1] Es wurden beide 4-BMS abgeschaltet.

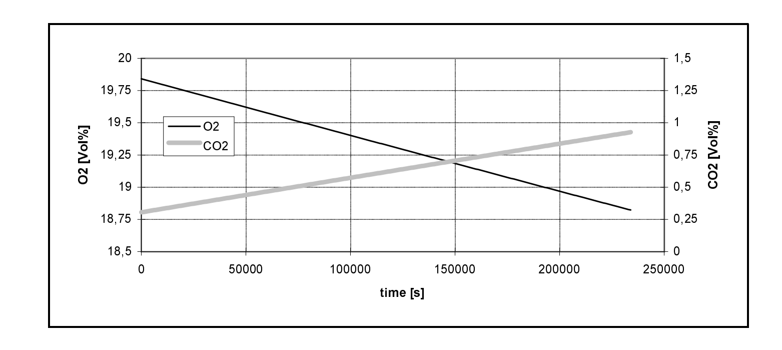


Abbildung 4.14.: Verlauf der Luftzusammensetzung bei ECLSS-Totalausfall

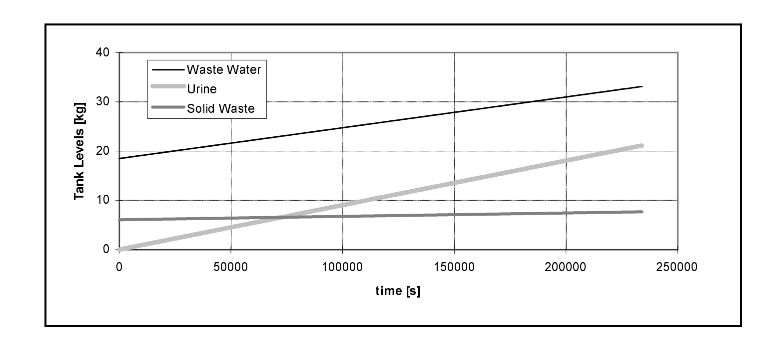


Abbildung 4.15.: Verlauf der Schmutztank-Inhalte bei ECLSS-Totalausfall

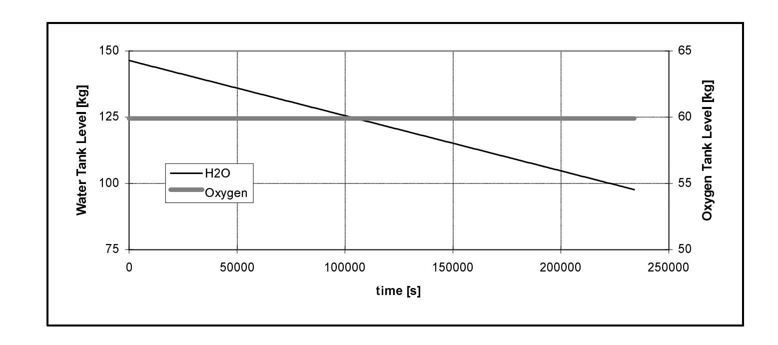


Abbildung 4.16.: Verlauf der Vorratstank-Inhalte bei ECLSS-Totalausfall

4.4.1. Langzeitsimulation

Das Anlaufverhalten des Systems, das in den Abbildungen 4.5, 4.6, 4.7 und 4.8 auf Seite 37 zu erkennen ist, zeigt ein Einpendeln des Kohlendioxyd-Gehaltes der Kabinenluft weit unterhalb des spezifizierten Grenzwertes. Die Form der Kurve resultiert aus dem konstanten Wirkungsgrad der als CO_2-Filter verwendeten Molekularsieb-Module, die erst ab einem bestimmten Kohlendioxyd-Gehalt der Luft die von der Crew eingebrachte Menge auch wieder entfernen und den CO_2-Anteil somit konstant halten können.

Luft

Der Sauerstoffgehalt der Luft sinkt zunächst aufgrund des konstanten Verbrauchs durch die Besatzung linear ab, bis die Ansprechschwelle des Sauerstoff-Regulators erreicht ist. Danach sorgt dieser für einen gleichbleibenden O_2-Anteil.

Der Energieverbrauch des ECLSS verläuft parallel zum Kohlendioxyd-Gehalt, da der Leistungsbedarf des Molekularsiebs als proportional zu der zu entfernenden CO_2-Menge modelliert wurde und die anderen Komponenten eine konstante Leistungsaufnahme haben. Der Gesamtverbrauch von etwas über 1,7 kW für die in die Modellierung einbezogenen Komponenten liegt etwas unter dem tatsächlich zu erwartenden Wert. Dies rührt daher, daß der Energieverbrauch der Luftentfeuchter und der Kabinenzirkulations-Ventilatoren mangels Daten nicht berücksichtigt werden konnte.

Energie

Die Verlaufskurven zeigen ein leichtes Ansteigen des Inhaltes des Schmutzwasser-Tanks, das durch die Unterdimensionierung des Wasserfilters bedingt ist. Alles von der Crew ausgeschiedene Urin wird vom Destillationsgerät weiterverarbeitet, so daß der Urintank leer bleibt. Der Fäkalientank füllt sich entsprechend der natürlichen Rate.

Tanks

In Abbildung 4.8 ist zu sehen, daß die Elektrolyseeinheit (SFWE) mehr Sauerstoff als nötig erzeugt, denn selbst nach dem Einschalten des Sauerstoffregulators nimmt der Sauerstofftankinhalt noch leicht zu. Dies wird durch die Annahme einer konstanten Produktionsrate im Modell des SFWE-Moduls verursacht. Der Frischwasservorrat nimmt hingegen langsam ab, da für die Elektrolyse mehr Wasser als nötig entnommen wird und das Wasserfilter, wie erwähnt, nicht alles anfallende Schmutzwasser verarbeiten kann.

Die Abbildungen 4.9 und 4.10 zeigen, daß der Inhalt aller Tanks bis auf den Frischwassertank während der simulierten 138 Tage zunimmt, während sich die Luftzusammensetzung auf den oben erwähnten stabilen Werten einpendelt. Die Zunahme an Gesamtmasse ist darauf zurückzuführen, daß die Nahrung für die Crew nicht mitbilanziert wurde.

Endstand

4.4.2. Ausfalluntersuchungen

Ohne Wiederverwendung des verschmutzten Wassers ist in Abbildung 4.12 eine rasche Entleerung des Trinkwassertanks sichtbar. Gleichzeitig füllt sich der Schmutzwassertank, wobei durch das automatische Überbordlassen von Schmutzwasser (*venting*) der Höchststand von 80 kg nicht überschritten wird. Die Untersuchung zeigt, daß das Wasserfilter ein kritisches Element des Raumstations-ECLSS ist. Jedoch bleibt bei Ausfall genügend Zeit für Gegenmaßnahmen, wie Reparatur und zeitweilige Reduktion des

Wasserfilter

Wasserverbrauchs. Gegebenenfalls kann auch der Urindestillator dazu verwendet werden, regeneriertes Wasser für Anwendungen mit geringeren Reinheitsanforderungen zu erzeugen.

Molekularsieb

Der Ausfall der Molekularsiebe führt zu einem konstanten Anstieg des Kohlendioxydgehalts in der Luft. Nach ca. 89 h ist der Grenzwert für dieses Gas erreicht. Die Crew hat auch in diesem Fall noch ausreichend Zeit für Gegenmaßnahmen wie Reparatur und Einsatz von Lithiumhydroxyd-Kanistern.

Totalausfall

Für den Fall des Versagens des gesamten ECLSS zeigt Abbildung 4.14 ein Ansteigen der Kohlendioxyd-Konzentration bei gleichzeitigem Absinken des Sauerstoffgehaltes. Letzteres führt auch zur ersten Grenzwertverletzung nach ca. 65 h. Würde dieser Zeitpunkt durch manuelle Zufuhr von Sauerstoff aus dem Tank hinausgezögert, so wäre kurze Zeit später auch der Grenzwert für den CO_2-Gehalt überschritten; aufgrund der erhöhten Crewanzahl deutlich früher als im vorhergehenden Fall. Die Tankinhalte steigen bzw. sinken entsprechend der natürlichen Produktions- oder Verbrauchsrate. Allein der Inhalt des Sauerstofftanks bleibt aufgrund der Ausfälle von Elektrolysegerät und Sauerstoffregulator konstant. Selbst bei Totalausfall – ausreichende Luftzirkulation vorausgesetzt – hat die Crew den Ergebnissen zufolge noch genügend Zeit für Gegenmaßnahmen oder im äußersten Fall zum geordneten Verlassen der Station.

5. Anwendung im Rahmen des Space Station Design Workshops

Erste Erfahrungen mit der Anwendung von MELISSA in der Lehre konnten im Rahmen des Space Station Design Workshops gemacht werden, der vom 17. bis 21. März 1997 am Institut für Raumfahrtsysteme stattfand. Die Teilnehmer dieser Veranstaltung waren Vertiefungsstudenten aus der Fachrichtung "Raumfahrtsysteme" der Universität Stuttgart, der École Nationale Supérieure de l'Aéronautique et de l'Espace und der École Nationale Supérieure d'Ingénieurs de Constructions Aéronautiques, beide in Toulouse, Frankreich.

Nach anfänglichen Einführungsvorlesungen, die die Teilnehmer mit der Thematik und Entwurfsmethodik vertraut machten, wurden drei Teams gebildet, die – betreut von Institutsmitarbeitern – selbständig eine Raumstation inclusive der Subsysteme Energie- und Lebenserhaltungssystem konzipierten und ihre Entwürfe unter Zuhilfenahme der vorhandenen Simulationsprogramme (GISSAD, IRIS, MELISSA) analysierten.

5.1. Einsatz von Melissa durch Workshop-Teilnehmer

Modellierung

Die ECLSS-Spezialisten der Teams wurden anhand der Einführung im Benutzerhandbuch (siehe Anhang E ab Seite 247) mit MELISSA vertraut gemacht. Das dabei entsprechend der Anleitung Schritt für Schritt erstellte Lebenserhaltungssystem mit geschlossenem Wasserkreislauf (Example.vi, siehe Anhang C ab Seite 91) diente als Basis für das zu entwerfende ECLSS. Der in Abbildung 5.1 dargestellte Entwurf eines Teams zeigt exemplarisch, wie durch Hinzufügen einiger Luftversorgungs-Module (Kabinensimulation, Sauerstoffregulator, Luftentfeuchter/Wärmetauscher, Molekularsieb, Spurengasfilter) der Luftkreislauf auf einfache Weise geschlossen werden konnte. Zusätzlich wurde noch die Gesamtmasse des Systems berechnet.

Simulation

Nach erfolgreicher Modellierung machten sich die Teams an die Bearbeitung der Simulationsaufgaben. Gefordert war, die Massenströme im regulären Betrieb zu ermitteln. Außerdem sollte die Reaktion des ECLSS auf Geräteausfälle untersucht werden. Als zusätzliche Hilfsmittel standen dazu detaillierte Handlungsanweisungen ("Kochrezepte") zur Verfügung. Diese Aufgaben konnten von den Teilnehmern ebenfalls ohne

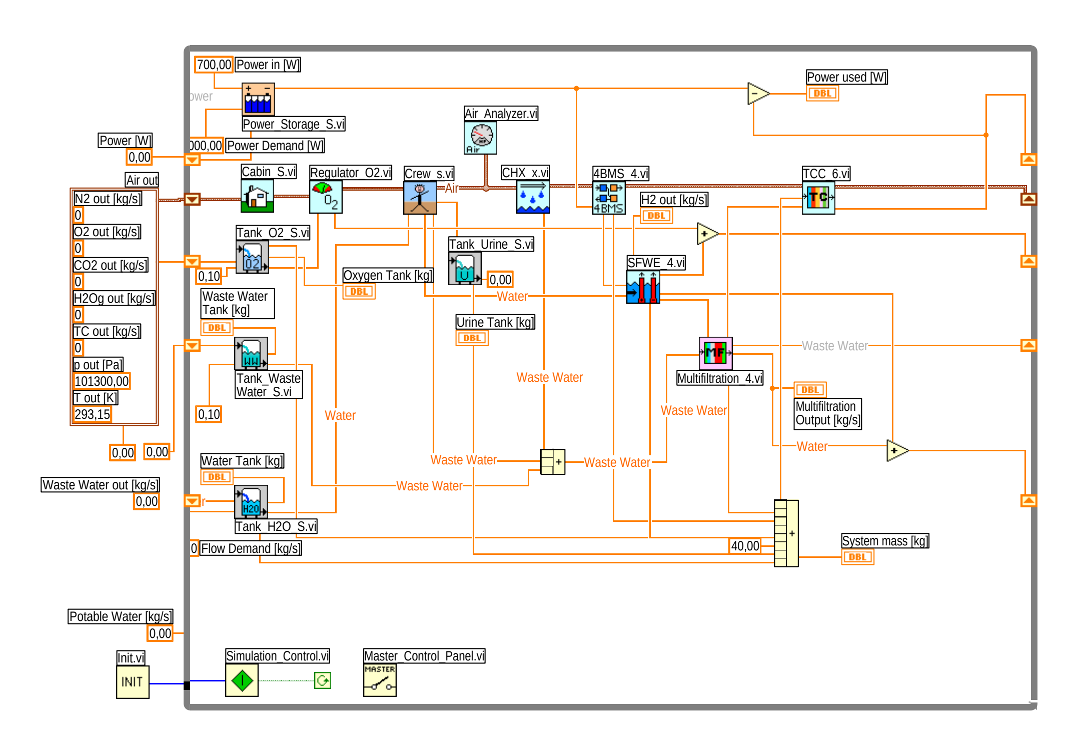


Abbildung 5.1.: ECLSS-Entwurf eines SSDW-Teams

größere Schwierigkeiten gelöst werden; jedoch war aufgrund der kurzen Einarbeitungszeit ständige Betreuerpräsenz unabdingbar.

5.2. Gewonnene Erfahrungen

Der erfolgreiche erste Einsatz von MELISSA in der Lehre bestätigte die Verwendbarkeit des gewählten Ansatzes für Modellierung und Simulation. Die vorhandenen erklärungsbedürftigen Abweichungen von der verwendeten Analogie, wie sie in Kapitel 2 beschrieben und erläutert werden, stellten nach entsprechenden Hinweisen und kurzer Einarbeitung kein Hindernis für den Umgang der Teilnehmer mit dem Programm dar.

Beurteilung

Auch die gewählte Form der Benutzerschnittstelle mit Meßinstrumenten und Anzeigen wurde von den Teilnehmer positiv beurteilt. Durch den hohen Interaktionsgrad und die dynamische Simulation konnte darüber hinaus auf eine die Benutzer motivierende Art ein vertieftes Verständnis für die Bestandteile und Zusammenhänge eines Lebenserhaltungssystems herbeigeführt werden.

Seitens der Teilnehmer kam der Wunsch auf, durch zusätzlich zur Verfügung gestellte Module des Lebenserhaltungssystems die Möglichkeiten des Designers beim Entwurf zu erweitern. Im folgenden abschließenden Kapitel wird darauf näher eingegangen werden.

6. Zusammenfassung

6.1. Lösungsansatz und Entwicklungsstand

Aufgabenstellung

Zum Vorentwurf von Raumstationen ist neben Werkzeugen für Geometrie- und Konfigurationsdefinition, wie sie z. B. von den bisherigen Elementen des *Space Station Design Workshop* zur Verfügung gestellt werden, auch ein geeignetes Programm zum Entwurf und zur Simulation von Subsystemen erforderlich. Dieses muß knappen Vorgaben und unterschiedlichen Beschreibungstiefen gerecht werden. Neben seiner Verwendung für Entwurfsaufgaben soll es auch in der Lehre zur Demonstration vernetzter Systemdynamik eingesetzt werden.

Lösungsansatz

Die auf der Basis der graphischen Programmierumgebung LabVIEW entwickelte Lösung MELISSA erlaubt die Modellierung und Simulation von vernetzten dynamischen Systemen mittels graphischem Netzwerkeditor und interaktiven Bedienelementen. Folgende Schritte sind dazu erforderlich:

1. Auswahl der benötigten Subsystemkomponenten
2. Verbinden der Ein- und Ausgänge der gewählten Komponenten in schaltplanähnlicher Form
3. Anschluß von virtuellen Anzeigeinstrumenten für Zwischen- und Endergebnisse
4. Einstellen der gewünschten Anfangsbedingungen der Komponenten
5. Durchführung der Simulationsrechnungen mit dem modellierten System
6. gegebenenfalls zusätzliche Auswertung der gewonnenen Daten mit externen Programmen

Für den Bereich des Lebenserhaltungs- und Energieversorgungssystems sind bereits umfangreiche Modulbibliotheken vorprogrammiert, die existierende Subsystem-Hardware modellieren. Weitere Geräte können nach dem Beispiel der vorhandenen einfach hinzugefügt werden.

Beispielanwendungen

Zum Nachweis der Leistungsfähigkeit von MELISSA wurde das Lebenserhaltungssystem der Internationalen Raumstation modelliert und sowohl unter Normalbedingungen als

auch bei Ausfall von Geräten untersucht. Ferner wurde MELISSA während eines Workshops für Raumstationsentwurf im Rahmen der Lehre eingesetzt.

Quantifizierung

In der derzeitigen Version erfüllt MELISSA im wesentlichen die Anforderungen der Aufgabenstellung. Zur Quantifizierung dieser Aussage und zur Identifizierung von Schwerpunkten für die zukünftige Weiterentwicklung wird die in Kapitel 2 hergeleitete *House of Quality*-Matrix verwendet (Abbildung 6.1). In der untersten Tabellenzeile ist bewertet, inwieweit MELISSA nach den gemachten Anwendungserfahrungen die unter "Umsetzung" angegebenen Anforderungen erfüllt. Dabei wird folgendes Schema verwendet:

- "erfüllt Anforderung in besonderem Maße" (+, 5 Punkte)
- "erfüllt Anforderung" (✓, 4 Punkte)
- "erfüllt Anforderung nur teilweise" (∼, 2 Punkte)
- "erfüllt Anforderung nicht" (–, 0 Punkte)

Auswertung

Mit Hilfe der Matrix kann für jede der fünf Kundenforderungen ermittelt werden, inwieweit MELISSA diese befriedigt. Dazu wird zunächst für jede Zelle der Matrix der in der zugehörigen Spalte vergebene Bewertungs-Punktwert mit einem den Zusammenhang wiedergebenden Faktor multipliziert. Ein starker Zusammenhang (◆) geht dabei mit dem Faktor 5 ein, ein schwacher Zusammenhang (○) mit dem Faktor 2. Die so berechneten Zellenwerte werden zeilenweise addiert und ergeben die Ist-Punktzahlen für jede der Forderungen. Als Referenz werden die Soll-Punktzahlen ermittelt, indem der beschriebene Rechengang mit 4 Bewertungspunkten für jede Umsetzungs-Spalte (entspricht "erfüllt Anforderung") wiederholt wird.

Ergebnis

Abbildung 6.2 zeigt das jeweilige Verhältnis von Ist- zu Sollwert für jede der fünf Kundenforderungen, aufgetragen als "Radar"-Diagramm gemäß [16]. Die ausreichende Erfüllung von vier der fünf Forderungen ist ersichtlich, genauso wie der Verbesserungsbedarf im Bereich der Einbindung in die vorhandene SSDW-Umgebung. Abschnitt 6.3 am Ende dieses Kapitels geht hierauf noch genauer ein.

6.2. Verbleibende Probleme

Trotz der insgesamt zufriedenstellenden Funktion der derzeitigen MELISSA-Version fielen im Laufe der Entwicklung und während der ersten Anwendungen einige Verhaltensweisen des Programmes auf, die im Rahmen weiterer Arbeiten beseitigt werden sollten.

Speichermodule

Die meisten dieser Probleme werden durch den zeitdiskreten Simulationsansatz verursacht und treten im Zusammenhang mit dem "Generic Storage"-Modul auf, das den Tanks und dem Energiespeicher zugrunde liegt. Am auffallendsten ist die stets in den Speichern verbleibende Restmenge. Der Speicheralgorithmus liefert jedoch auch ungenaue Ergebnisse für einen Tankinhalt, wenn der Benutzer bei laufender Simulation und

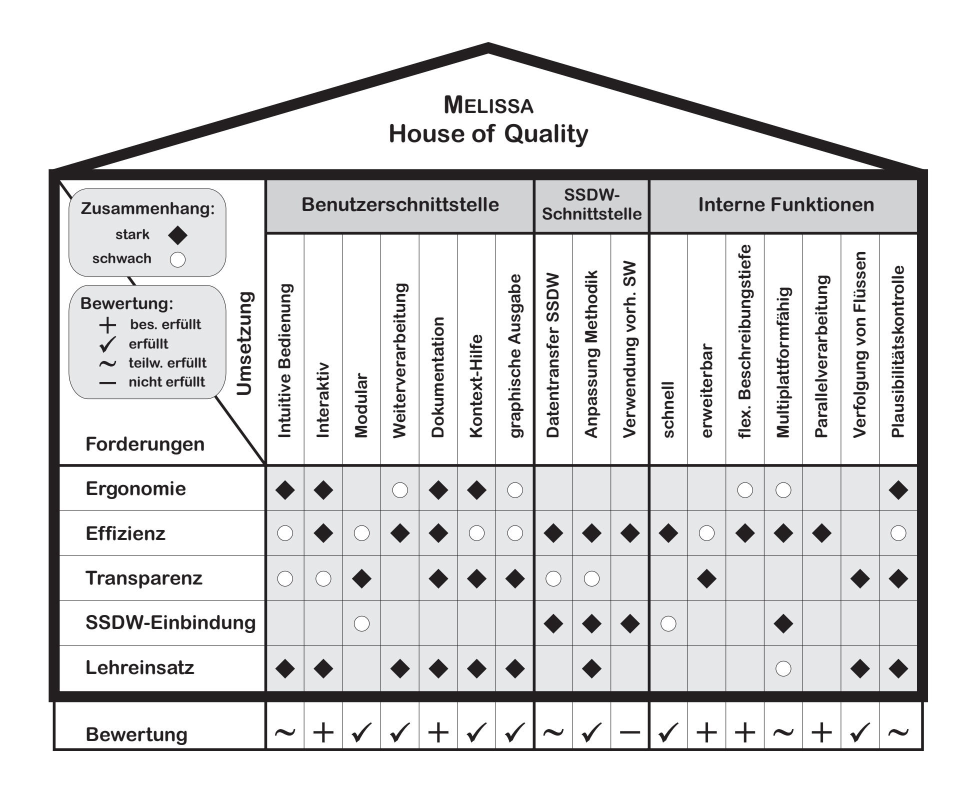


Abbildung 6.1.: Die *House of Quality*-Matrix mit den Bewertungen

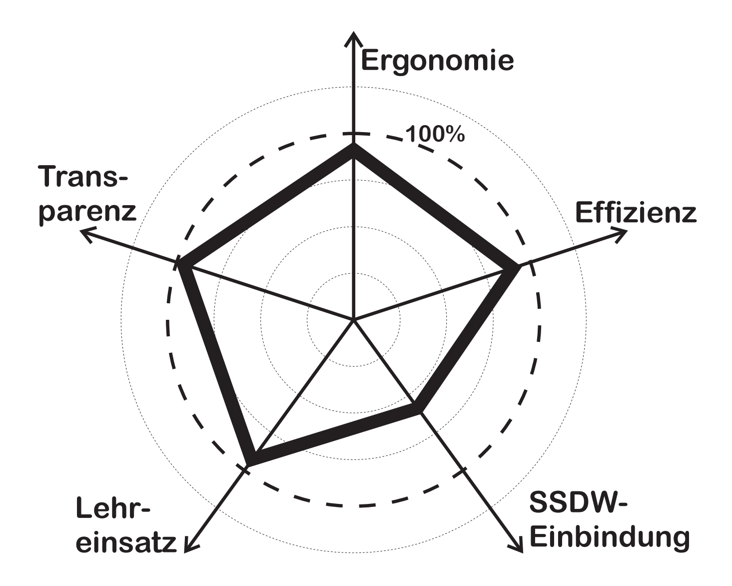


Abbildung 6.2.: Erfüllung der Kundenforderungen durch MELISSA

bis auf die Restmenge leerem Tank den Simulations-Zeitschritt (*time step*) ändert. Hier kann gegebenenfalls eine Modifikation der Speicherlogik Abhilfe schaffen.

Sauerstoffregulator

Ein weiteres kritisches Element ist der Sauerstoffregulator. Der verwendete Algorithmus umgeht bei der Berechnung der Sauerstoffmenge, die zum Erreichen des vorgegebenen O_2-Partialdruckes dem Luftstrom zugefügt werden muß, eine eigentlich notwendige numerische Iteration durch Verwendung einer linearen Approximation. Dadurch wird zwar erhebliche Rechenzeit eingespart, jedoch ist der dabei verwendete lineare Faktor für "normale" Luftzusammensetzungen, -mengen und -drücke optimiert. Es kann also vorkommen, daß der Sauerstoffgehalt des austretenden Luftstromes trotz vorhandener Sauerstoffversorgung niedriger als der vorgegebene Wert ist. Dieses Fehlverhalten kann durch weitere Optimierung der Approximation verringert oder durch Verwendung des Iterationsalgorithmus (auf Kosten der Ausführungsgeschwindigkeit) verhindert werden.

LabVIEW

Einige Schwierigkeiten werden direkt durch die Verwendung von LabVIEW verursacht. So muß sorgfältig darauf geachtet werden, Unterprogramme mit lokaler Datenspeicherung pro Diagramm nur einmal einzusetzen. Dies trifft z. B. auf die diversen Tanks oder das Energiespeicher-Modul zu. Das Problem rührt daher, daß LabVIEW jedes Unterprogramm samt dazugehörigen lokalen Variablen und Front Panel-Werten nur einmal im Speicher hält, auch wenn es mehrfach in einem Diagramm vorkommt.

Mehrfach-VIs

Falls der Benutzer etwa ein System mit zwei Wassertanks simulieren möchte und daher einfach das "Tank_H20"-Symbol zweimal auf das Diagramm zieht, würde LabVIEW trotzdem nur einen Satz lokale Variablen für den Tankinhalt etc. anlegen. Das Entnehmen von Wasser aus einem Tank würde sich also genauso im anderen niederschlagen, und die Änderung der Tankkapazität des einen würde auch den anderen Tank verändern. Als Ausweg aus dieser Situation bietet sich an, mehrfach verwendete Module verschieden zu benennen (z. B. "Tank_H2O_1" und "Tank_H2O_2"), worauf LabVIEW dann die VIs als separate Programme behandelt und die oben geschilderte Problematik nicht mehr auftritt.

Zeit

Eine weitere kleine Unstimmigkeit kann sich bei der Berechnung von Simulationszeit und -datum ergeben, wenn während der Simulation die Simulationszeit die Grenze zwischen Winter- und Sommerzeit oder umgekehrt überschreitet. Die entsprechenden Änderungen werden von den verwendeten LabVIEW-Algorithmen automatisch durchgeführt, so daß eventuell eine Stunde zuviel oder zuwenig angezeigt wird. Hier beseitigt ein Vergleich der angezeigten Simulationszeit mit den gezählten verstrichenen Sekunden seit Simulationsanfang (Anzeige *Simulation Seconds* im Time Control Panel) eventuelle Unsicherheiten.

Besonderheiten

Außerdem kann im Rahmen von weiterführenden Arbeiten versucht werden, die in Abschnitt 3.2.4 angesprochenen erklärungsbedürftigen Besonderheiten in ihrer Anzahl oder Tragweite zu reduzieren, um die intuitive Bedienbarkeit von MELISSA zu erhöhen.

6.3. Ausblick

Eine Übersicht über geplante oder gewünschte Erweiterungen und Verbesserungen von Melissa schließt den Hauptteil dieses Berichtes ab. Diese Übersicht bietet neben der obigen Aufstellung von verbesserungsbedürftigen Funktionen einen Ausblick auf mögliche weiterführende Arbeiten. Zusätzlich steht noch die Verifikation von Modul- und Programmfunktionalität durch Vergleich der Ergebnisse von Simulationsrechnungen mit tatsächlichen Betriebsdaten von Mir oder ISS aus. Dies ist jedoch erst nach Veröffentlichung entsprechender Daten möglich.

Erweiterung des Simulationsumfangs:

- Abspeicherungs- und Lademöglichkeit für Simulationsparameter wie Luftzusammensetzung, Tankinhalte, Initialisierungswerte
- Integration des Thermalkontrollsystems
- Einbeziehung von Kosten, Zuverlässigkeit, Wartungsaspekten, Lebensdauer

Programmierung zusätzlicher Module:

- Nahrungsversorgung (Lagertanks, Produktionseinrichtungen)
- Bahnregelung (elektrisch, Kaltgas, H_2/O_2)
- Luftkreislauf: EDC, SAWD, APC, LiOH, Bosch, Superoxyde, WVE
- Wasserkreislauf: TIMES, VAPCAR, Verdunster, Umkehrosmose

Verbesserung vorhandener Module:

- verstärkter Abfang von fehlerhaften Eingaben, vor allem in den VIs der Hilfs-Bibliotheken
- anfängliche Initialisierung der Simulation nur bei Bedarf
- Vent und Shunt mit Hysterese
- Vereinfachung der Datenspeicherung (Log_Data.vi), z. B. durch Einsatz von globalen Variablen
- Übernahme von Bahndaten aus IRIS
- Übernahme von Konfigurationsdaten aus GISSAD
- variabler Kabinendruck; Leckagesimulation
- druck- und temperaturabhängige Prozesse in den Modulen des Luftkreislaufs
- temperaturabhängige Funktion des Radiator-Moduls

- Kühleffekt des CHX-Moduls; Lufterwärmung durch Crew, Geräte, Sonneneinstrahlung
- variable Verbrauchs- und Produktionsraten der Crew (abhängig von Eingabeflüssen)
- Dichteberechnung abhängig von Druck und Temperatur
- selbständiges Abschalten von Geräten bei fehlender Energie- und Massenversorgung

Zuerst sollte die Programmierung zusätzlicher Module in Angriff genommen werden, um vergleichende Untersuchungen zu ermöglichen. Auch eine genauere Modellierung des Einflusses der Kabinenatmosphäre wäre wünschenswert. Schließlich würde die Erweiterung der Datenspeicherungsfunktionalität die Durchführung von Simulationen mit MELISSA vereinfachen.

Literaturverzeichnis

[1] Aspen Plus Information. Aspen Technology, Inc. Cambridge, Mass. Web-Veröffentlichung, 1. Mai 1996. http://www.aspentec.com/products/software/aspenplus/aspenbro.html

[2] CASE/A: Computer Aided System Engineering and Analysis. NASA Marshall Space Flight Center, Web-Veröffentlichung, ohne Datum. http://cognac.cosmic.uga.edu/abstracts/mfs-28573.html (Stand 06. April 1997).

[3] Crutcher, A.; Perez Vara, R.: ECOSIM – Environmental Control & Life Support Simulation.

[4] Dubbel Taschenbuch für den Maschinenbau. 17. Auflage. Beitz, W.; Küttner, K.-H. (Hrsg.). Springer Verlag, Berlin 1990.

[5] Eckart, Peter: Spaceflight Life Support and Biospherics. Microcosm Press, Torrance CA 1996.

[6] Fabig, Joachim: Bewertung von Konzepten zur regenerativen Stoffwirtschaft für Raumstationen. Studienarbeit IRS-94-S-03. Institut für Raumfahrtsysteme, Universität Stuttgart, 1994.

[7] Feustel-Büechl, Jörg: Europa braucht die Internationale Raumstation. S. 1-3, DGLR-Mitteilungen 3/95. Deutsche Gesellschaft für Luft- und Raumfahrt, Bonn 1995.

[8] Hauser, John R.; Clausing, Don: The House of Quality. Harvard Business Review, May-June 1988, S. 63 ff.

[9] ISS Technical Data Book: Sub-System Environmental Control and Life Support System (ECLSS) on Flight 19A. NASA Web-Veröffentlichung 20. März 1995. http://issa-www.jsc.nasa.gov/ss/techdata/ECLSS/19A.html

[10] Johnson, Gary W.; Kay, Meg F.: LabView Programmer's Style Guide. Web-Veröffentlichung, 14. Juli 1994. Verfügbar als "lvstyle.ps" unter ftp://ftp.natinst.com/support/labview/documents/style-guide/postscript/

[11] Kurtz, H.: Energieversorgungssysteme für die Raumfahrt. Vorlesungsskript. Institut für Raumfahrtsysteme, Universität Stuttgart, 1992.

[12] LabView User Manual for Windows. National Instruments, Austin TX 1994.

[13] Leonard, Kathleen M.; Aksamentov, Valery K.: Compatibility of Russian Life-Support Systems to Future Space Missions. Journal of Aerospace Engineering, Vol. 8, No. 1, Januar 1995. S. 68 ff.

[14] Marquardt, W.: Trends in Computer-Aided Process Modeling. Computers and Chemical Engineering, Vol. 20, No. 6/7, Juni / Juli 1996. S. 591 ff.

[15] Mavris, Dimitri N.: HSCT Design Study Tools. Vortrag, Georgia Tech AE 6351. Georgia Institute of Technology, Atlanta, GA 1995.

[16] Mavris, Dimitri N.; DeLaurentis, Daniel: An Integrated Approach to Military Aircraft Selection and Concept Evaluation. Paper presented at the 1st AIAA Aircraft Engineering, Technology and Operations Congress, Los Angeles, CA, September 19-21, 1995. American Institute of Aeronautics and Astronautics, 1995.

[17] Messerschmid, Ernst; Bertrand, Reinhold; Pohlemann, Frank: Raumstationen (Vorabversion). Springer Verlag, Heidelberg 1997.

[18] Mir Status Report. NASA Web-Veröffentlichung, 28. März 1997. http://shuttle-mir.nasa.gov/shuttle-mir/mir23/status/week4/missrpt.html

[19] Mir Status Report. NASA Web-Veröffentlichung, 4. April 1997. http://shuttle-mir.nasa.gov/shuttle-mir/mir23/status/week5/missrpt.html

[20] Newman, William M.; Lamming, Michael G.: Interactive System Design. Addison-Wesley Publishing Company, Cambridge 1995.

[21] Niven, Larry; Pournelle, Jerry: Lucifer's Hammer. Ballantine Books. New York, 1977. Zitat auf S. 638.

[22] Osburg, Jan: Space Missions to Mars - Past, Present and Future. Georgia Tech AE 6250 Term Paper. Georgia Institute of Technology, Atlanta, GA, 1996.

[23] Puttkamer, Jesco v.: Jahrtausendprojekt Mars. Albert Langen / Georg Müller Verlag. München 1996.

[24] Sanchez, Susan M. et al.: Quality by Design. In: Kusiak, Andrew (Hrsg.): Concurrent Engineering – Automation, Tools, and Techniques. John Wiley & Sons, New York 1993.

[25] Schad, Ryan C.: Don't Let Recycle Streams Stymie Your Simulations. Chemical Engineering Progress, Dezember 1994, S. 68 ff.

[26] Schuler, Hans (Hrsg.): Prozeßsimulation. VCH, Weinheim 1995.

[27] Simulink User's Guide. The MathWorks, Inc. Natick, Mass. März 1992.

[28] Unterlagen und Notizen zur Vorlesung "Raumfahrtsysteme". Institut für Raumfahrtsysteme, Universität Stuttgart, Sommersemester 1993.

[29] Sullivan, L. P.: Quality Function Deployment. Quality Progress, June 1986, S. 39 ff.

[30] Taylor, Tony: Data Acquisition – One Step at a Time. Physics World vol. 8 no. 10, Oktober 1995, S. 49 - 53.

[31] Thimbleby, Harold: User Interface Design. Addison-Wesley, London 1990.

[32] Utilisation of the International Space Station. ESA Document No. MSM-4785. 22. Juni 1995.

[33] Wertz, James R.; Larson, Wiley, Larson J.: Space Mission Analysis and Design. Microcosm Inc. Torrance, CA 1992.

[34] Wieland, P.: Designing for Human Presence in Space. NASA Marshall Space Flight Center, Web-Veröffentlichung 1996. http://augusta.msfc.nasa.gov/ed61/papers/rp1324/intro.html (Stand 06. April 1997).

[35] Wydeven, Theodore: A Survey of Some Regenerative Physico-Chemical Life Support Technology. NASA Technical Memorandum 101004. Ames Research Center, Moffett Field CA, November 1988.

Auswahl weiterführender Literatur

[36] Belaventsev, Y; et al.: A System for Oxygen Generation from Water Electrolysis Aboard MIR. Proceedings of the 4th European Symposium on Space Environment Systems, ESA SP-324, Oktober 1991. S. 477 - 479.

[37] Doll, Susan; Tillotson, Brian: TRIALSS – Tool for Rapid and Intelligent Advanced Life Support System Selection and Sizing. Presented on the 22nd International Conference on Environmental Systems, Seattle, WA, July 13 - 16, 1992. SAE Technical Paper Series 921123. SAE International, Warrendale PA 1992.

[38] Gustavino, Stephen R. et al.: Concepts for Advanced Waste Water Processing Systems. SAE Technical Paper 941500. Presented on the 24th International Conference on Environmental Systems and 5th European Symposium on Space Environmental Control Systems, Friedrichshafen, Germany, June 20 - 23, 1994. SAE International, Warrendale PA, 1994.

[39] Hall, John B. Jr.; Thomas, Carolyn C.: Alternate Environmental Control and Life Support Systems Technologies for Space Station Application. SAE Technical Paper 941464. Presented on the 24th International Conference on Environmental Systems and 5th European Symposium on Space Environmental Control Systems, Friedrichshafen, Germany, June 20 - 23, 1994. SAE International, Warrendale PA, 1994.

[40] Hutchens, Cindy F.: A Description and Comparison of US and Russian Urine Processing Hardware for the International Space Station. SAE Technical Paper 941251. Presented on the 24th International Conference on Environmental Systems and 5th European Symposium on Space Environmental Control Systems, Friedrichshafen, Germany, June 20 - 23, 1994. SAE International, Warrendale PA, 1994.

[41] Life Support and Habitability Manual. ESA PSS-03-406, 1992.

[42] Samsonov, N. M. et al.: Development and Operation of the MIR Space Station System for Hygiene Waste Water Recovery. SAE Technical Paper 941535. Presented on the 24th International Conference on Environmental Systems and 5th European Symposium on Space Environmental Control Systems, Friedrichshafen, Germany, June 20 - 23, 1994. SAE International, Warrendale PA, 1994.

[43] Sulzman, Frank (Hrsg.): Life Support and Habitability. Space Biology and Medicine Series, Volume II. American Institute of Aeronautics and Astronautics, Washington, DC, 1993.

A. Glossar und Abkürzungsverzeichnis

Nachfolgend werden die im Text verwendeten Abkürzungen und Begriffe erläutert. Einige der Definitionen sind Übersetzungen aus [12] und [5]. Dort sind auch weitere LabView- und ECLS-spezifische Begriffe erklärt.

4-BMS: Vierbett-Molekularsieb (zur Extraktion von CO_2 aus der Atemluft).

AOCS: Attitude and Orbit Control System (Lage- und Bahnregelungssystem).

APC: Air Polarized Concentrator (ebenfalls zur Extraktion von CO_2 aus der Atemluft).

Bibliothek: Eine Sammlung von zusammenhängenden VIs.

CHX: Condensing Heat Exchanger (Wärmetauscher / Luftentfeuchter).

Cluster: Eine geordnete, nicht indizierte Gruppe von Instrumenten beliebigen Typs. Die Elemente müssen alle Controls oder alle Indicators sein.

Connector: Der Teil eines VIs, der die Ein- und Ausgabeschnittstellen bereitstellt.

Control: Ein Front-Panel-Objekt, mit dem man Daten eingeben oder angezeigte Daten ändern kann.

Diagramm: Das weiße Fenster eines VIs, in dem der Algorithmus des Programmes mittels der von LabView zur Verfügung gestellten graphischen Programmiersprache "G" eingegeben wird.

Draht: Eine symbolische Datenverbindung zwischen einer Datenquelle und einer Datensenke.

ECLS: Environmental Control and Life Support (Lebenserhaltung).

ECLSS: Environmental Control and Life Support System (Lebenserhaltungssystem).

EDC: Electrochemical Depolarized Concentration (Methode zur CO_2-Entfernung aus der Luft).

EPS: Electrical Power Supply System (Energieversorgungssystem).

Flow: Siehe "Fluß".

Fluß: Ein Wert, der eine Änderung pro Zeit repräsentiert, z. B. ein Massenfluß in [kg/s] oder Leistung (Energie pro Zeit) in [W].

Front panel: Das graue Fenster eines VIs, in dem Instrumente als Schnittstellen zum Benutzer und zu anderen Programmen zur Verfügung stehen.

Function: Ein eingebauter Befehl, der meist eine einfache Berechnung oder Aktion ausführt (z. B. "plus", "dividiert durch", ...).

Global variable: Eine Variable, auf die von allen im Speicher befindlichen VIs her zugegriffen werden kann. Sie werden von LabVIEW als spezielles VI gespeichert, das nur ein Front Panel und kein Diagramm besitzt. Globale Variablen stehen im "Numeric" Untermenu der Functions Palette zur Verfügung.

Icon: Das graphische Symbol für ein VI.

Indicator: Ein Front-Panel-Element, auf dem Daten angezeigt werden. Eingaben sind nicht möglich.

Instrument: Ein Front-Panel-Objekt, das Daten enthält (d. h. ein Control oder ein Indicator).

ISS: International Space Station.

LabView: Name des als Basis für MELISSA verwendeten Programms (Laboratory Virtual Instrument Engineering Workbench). Hersteller: National Instruments Inc., Austin, Texas.

Library: Siehe "Bibliothek".

LiOH: Lithiumhydroxyd (zur nicht-regenerativen Entfernung von CO_2 aus der Atemluft).

MELISSA: Name der entwickelten Simulationsumgebung (*M*odular *E*nvironment for *Li*fe-Support *S*ystems *S*imulation and *A*nalysis).

Modul: Ein VI, das eine Baugruppe eines Systems simuliert.

Palette: LabVIEW-Ausdruck für ein graphisches Menü, dessen Elemente durch Symbole dargestellt werden. Die am meisten benutzten Paletten sind die Controls Palette, die Functions Palette (beide auf Seite 13 abgebildet) sowie die Tools Palette.

SAWD: Solid Amine Water Desorption (Methode zur Entfernung von Kohlendioxyd aus der Luft).

SFWE: Static Feed Water Electrolysis (Elektrolysegerät zur Erzeugung von Sauerstoff und Wasserstoff aus Wasser).

Shift register: Ein optionales Element von Schleifenstrukturen, mit dem der Wert einer Variablen von einem Schleifenschritt in den nächsten weitergereicht wird.

SI: Internationals System der Maßeinheiten (Système International d'Unités), das in dieser Arbeit ausschließlich verwendet wird.

SSDW: Space Station Design Workshop (am Institut für Raumfahrtsysteme entwickelte Entwurfsumgebung für Raumstationen mit Modellierungs- und Simulationswerkzeugen).

SubVI: Ein VI, das eine Unteraufgabe in einem hierarchisch höherstehenden VI ausführt.

TCC: Trace Contaminant Control (Filter zur Entfernung von Spurengasen aus der Luft).

TIMES: Thermoelectric Integrated Membrane Evaporation System (Verfahren zur Regenerierung von Wasser).

Tool: Ein spezieller LabVIEW-Cursor, mit dem bestimmte Funktionen ausgeführt werden können. Beispiele sind die Tools zum Verschieben von Objekten oder zum Verdrahten. Die Tools befinden sich auf der auf Seite 13 dargestellten Tools Palette.

USOS: United States Orbital Segment. Bezeichnung für den amerikanischen Anteil der Internationalen Raumstation.

VAPCAR: Vapor Phase Catalytic Ammonia Removal (Physikalisch-chemisches Verfahren zur Wasserreinigung).

VCD: Vapor Compressed Distillation (Destillationsverfahren zur Wasser- und Urinaufbereitung).

VI: "Virtuelles Instrument", Bezeichnung für ein Programm in LabVIEW.

Wire: Siehe "Draht".

WVE: Water Vapor Electrolysis (Methode zur Erzeugung von Sauerstoff aus der Luftfeuchtigkeit).

B. Ergebnisse der ISS-Simulationen

In diesem Anhang sind die Ergebnisse der zur Anwendungsdemonstration von MELISSA verwendeten Simulationen in tabellarischer Form zusammengefaßt. Die Auswertung und graphische Darstellung erfolgt im Hauptteil der Diplomarbeit in Kapitel 4 ab Seite 31.

Die Reihenfolge der Tabellen ist:

- Anlaufverhalten (7 Blatt)
- Ausfall des Multifiltrationsgerätes (1 Blatt)
- Ausfall der Molekularsiebe (3 Blatt)
- Ausfall des gesamten ECLSS (3 Blatt)

Folgende Daten wurden während der Simulationsläufe gespeichert und sind in den Tabellen angegeben:

1. Simulationsschritt-Zähler
2. Verstrichene Sekunden seit Simulationsbeginn
3. Zusammensetzung der Luft in Volumen-%:
   - Stickstoff
   - Sauerstoff
   - Kohlendioxyd
   - Wasserdampf
   - Spurengase
4. verbrauchte Leistung in Watt
5. erzeugte Leistung in Watt
6. gespeicherte Energie in Joule

7. Tankstände in kg:
    - Schmutzwasser
    - Urin
    - Fäkalien
    - Trinkwasser
    - Sauerstoff
8. Simulationsdatum und -zeit

C. Dokumentation der entwickelten VIs

Auf den folgenden Seiten werden die im Rahmen dieser Diplomarbeit entwickelten VIs dokumentiert. Die Seiten wurden direkt mit der LabVIEW-internen Dokumentationsfunktion erzeugt. Die Ausdrucke sind entsprechend den MELISSA-Modulbibliotheken geordnet:

ECLS Modules: Hier findet der Benutzer alle VIs, die ECLSS- und EPS-spezifische Hardware simulieren.

ECLS Library: In dieser Bibliothek sind Module zusammengefaßt, die in einer Simulation ECLSS- und EPS-spezifische Hilfsfunktionen ausführen oder die Grundbausteine für die Module in der "ECLS Modules"-Library zur Verfügung stellen.

Auxiliary Library: Einige allgemeine Funktionen ohne direkten ECLSS- oder EPS-Bezug, die während der Erstellung von MELISSA programmiert wurden und für nachfolgende Benutzer von gewissem Nutzen sein können, werden in dieser Library zur Verfügung gestellt.

C.1. ECLS Modules

Module Name	*Purpose*	*Inputs*	*Outputs*	*Remarks*
4-BMS	Four-bed molecular sieve to remove CO_2 from an air stream.	air, power	CO_2, heat, unit mass, air, (power)	Sized for 4 people.
Air analyzer	Provides the mass and volume flow rates, the volumetric percentages and the partial pressure of an air stream.	air	-	
Cabin	Models the contributions of the pressurized volume.	air	air	Volume adjustable.
CHX	Condensing heat exchanger to extract water vapor from air.	air	air, water, heat	Unlimited processing capacity.
Crew	Models the contribution of the crew to the ECLS.	air, potable water, food	air, waste water, urine, feces, heat, (potable water, food)	Crew size and various comfort levels are selectable.
Master control panel	Allows convenient remote controlling of the ECLS modules.	-	-	Uses global variable "Master_Control".
Multifiltration	Multifiltration unit for conversion of waste water to potable water.	waste water, power	clean water, heat, unit mass, (waste water, power)	Sized for 4 people.
Power storage	Power supply and energy storage unit.	Charging power, returned power, power demand	power, energy level, estimated mass.	Capacity adjustable.
Radiator	Heat dump element.	heat	unit mass, (heat)	Simple model.
Regulator O_2	Oxygen partial pressure regulator.	air, oxygen	air, (oxygen)	Adjustable minimal, optimal and maximal O_2 partial pressures.
Sabatier	Transforms CO_2 to CH_4 and H_2O using additional H_2.	CO_2, H_2, power	CH_4, H_2O, heat, unit mass, (CO_2, H_2, power)	Sized for 5 people.
SFWE	Static Feed Water Electrolysis unit to generate oxygen from water.	water, power	oxygen, hydrogen, unit mass, (water, power)	Sized for 4 people.
Shunt	Element for converting power into heat (electric energy dump).	power, energy level, activation level, shunt rate	power, heat flow, activation flag, counter	
Solar array	For power generation.	solar flux, angle of incidence	power generated, unit mass	Size adjustable, cell type and age selectable.
Tanks	Various storage tanks for CH_4, CO_2, H_2, H_2O, N_2, O_2, solid waste, urine, waste water.	flow return, flow demand	outflow, tank level, empty mass	Capacity and initial level adjustable.
TCC	Trace Contaminant Control unit to remove trace contaminants from an air stream.	air, power	air, unit mass, (power)	Sized for 6 people.
VCD	Vapor Compressed Distillation unit to process waste water and urine.	waste water, power	clean water, brine, heat, unit mass, (waste water, power)	Sized for 6 people (waste water and urine processing) respectively 15 people (only urine processing).
Vent	Generic vent to space for getting rid of liquids.	mass flow, storage level, activation level, vent rate	venting indicator, vented mass counter, (mass flow)	

C.2. ECLS Library

Module Name	*Purpose*	*Inputs*	*Outputs*	*Remarks*
Air in / out	Custom control / indicator for an air stream, consisting of individual elements for its five species (N_2, O_2, CO_2, water vapor, trace contaminants), pressure and temperature.	-	-	
Eclipse simulation	Provides the available solar power flux density [W/m2] at any given time during a simulation.	altitude, inclination, Ω, Θ	solar flux, orbit time, eclipse time [s] and [%]	Adjustable to different planetary environments (Venus, Earth, Mars, Jupiter).
Generic storage	Base VI for power storage and tanks.	influx, flow return, capacity, initial level	flow out, level, overflow indicator	Not needed for simulation-level programming.
Init	Initializes simulation elements.	-	dummy	Place outside the simulation while loop and wire to it.
Initialize All	Global variable initialization flag.	-	-	Not needed for simulation-level programming.
Mass flow in / out	Custom control / indicator for mass.	-	-	
Master control	Global variable on/off flag cluster.	-	-	Not needed for simulation-level programming.
Partial pressure	Calculates the partial pressures of the five species in an air stream.	air	partial pressures	
Power in / out	Custom control / indicator for power	-	-	
Simulation control	Provides various functions for convenient simulation: pause, stop, initialize, call auxiliary panels (time control, log data, air analyzer, master control)	loop counter, paths, names	continue flag	Opens automatically when called.
Species accounting	Provides mass accounting for air species.	flow return, initial species mass, circulation demand	outflux, available species mass	Not needed for simulation-level programming.
Tank_S	Base module for the various tanks of the ECLS modules library.	tank type, flow return, capacity, initial level, flow demand	empty mass, level, flow out, overflow	Not needed for simulation-level programming.
Time	Global variable containing the time control variables.	-	-	
Time control	Interface for simulation time and time step control.	-	-	Not needed for simulation-level programming; is called by the Simulation control VI.
Vol_percent	Calculates the volumetric fractions of the species in an air stream.	air	volumetric fractions	
Volume flow	Calculates the individual and total volumetric flow rates of the species in an air stream.	air	flow rates	

C.3. Auxiliary Library

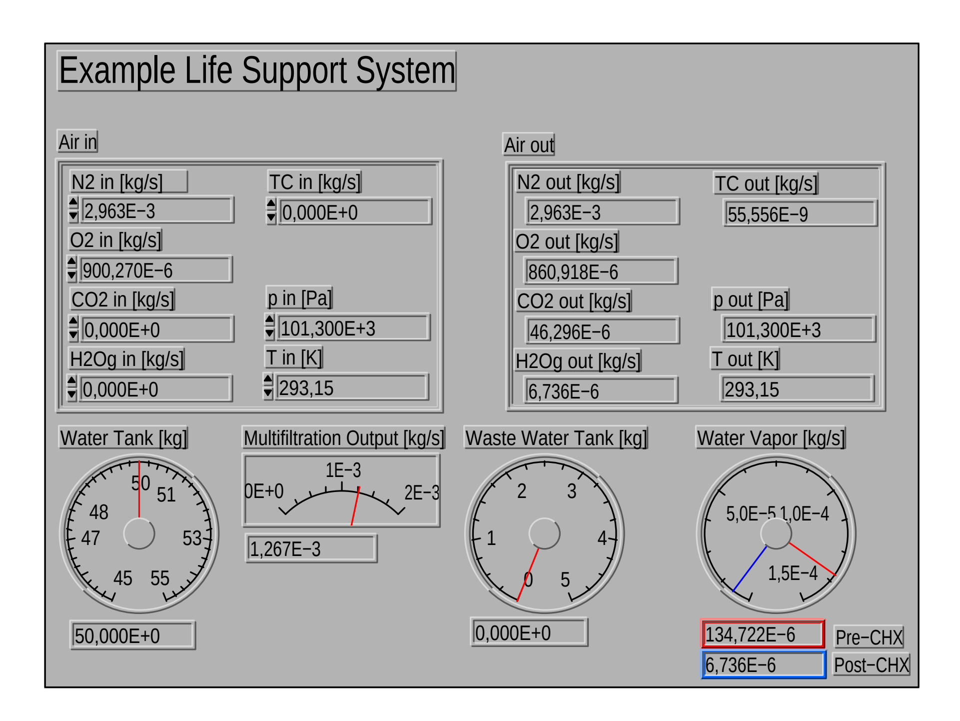
Module Name	*Purpose*	*Inputs*	*Outputs*	*Remarks*
Averaging	Calculates the average of the last N values of an input.	input value, N	average	
Deg_Rad	Converts degrees to radians and vice versa.	angle in deg (or rad)	angle in rad (or deg)	Can convert in both directions simultaneously and independently.
Density	Provides the densities of various species.	pressure, temperature	cluster containing densities	Output independend of pressure and temperature in the current implementation.
Exponential	Calculates x^y.	x, y	result	
Limiter_max	Limits a value to a given upper bound.	value, upper limit	limited value	
Limiter_min	Limits a value to a given lower bound.	value, lower limit	limited value	
Limiter_minmax	Limits a value to a given upper and lower bound.	value, upper and lower limit	limited value	
Log data	Saves simulation data in spreadsheet-readable format.	data array, data names, logfile name and path, simulation counter	-	
Remainder	Calculates the remainder of x/y.	x, y	remainder	$-1 < \text{remainder} < +1$
Show panel	Opens the front panel of a specified VI.	panel switch, panel name and path	-	
Sine array	Generates an array filled with sine values.	amplitude, period	array	Adapted from a LabView example.
Sine generator	Generates a sine wave.	amplitude, period	sine value	

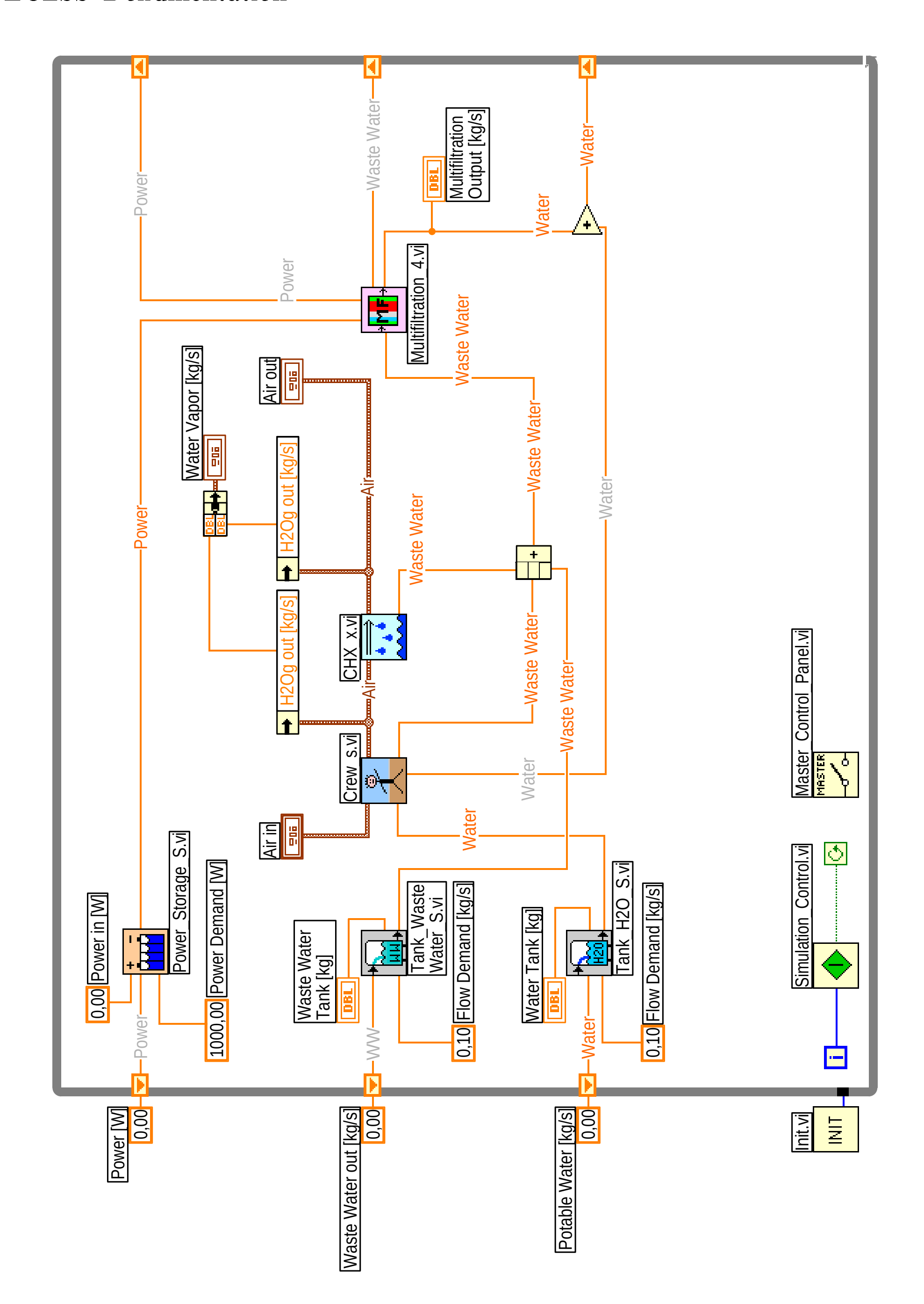
D. ECLSS–Dokumentation

Auf den folgenden Seiten werden die als Beispiele verwendeten VIs durch die von LabView erzeugten Ausdrucke dokumentiert. Die Dateinamen sind:

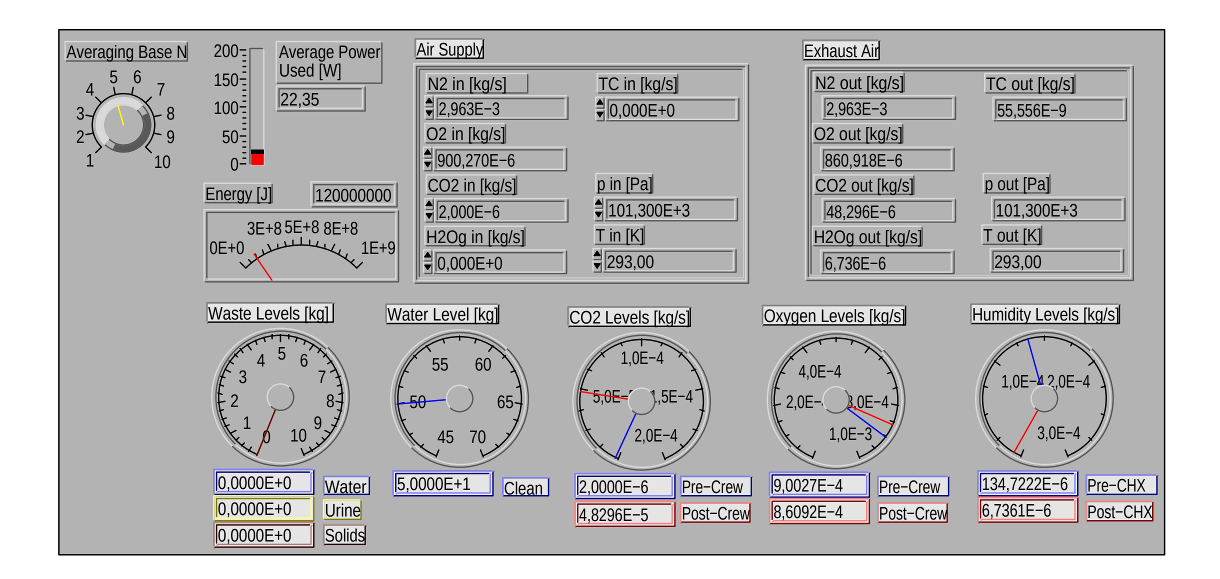
- Example.VI
- Water.VI
- ISS_US.VI

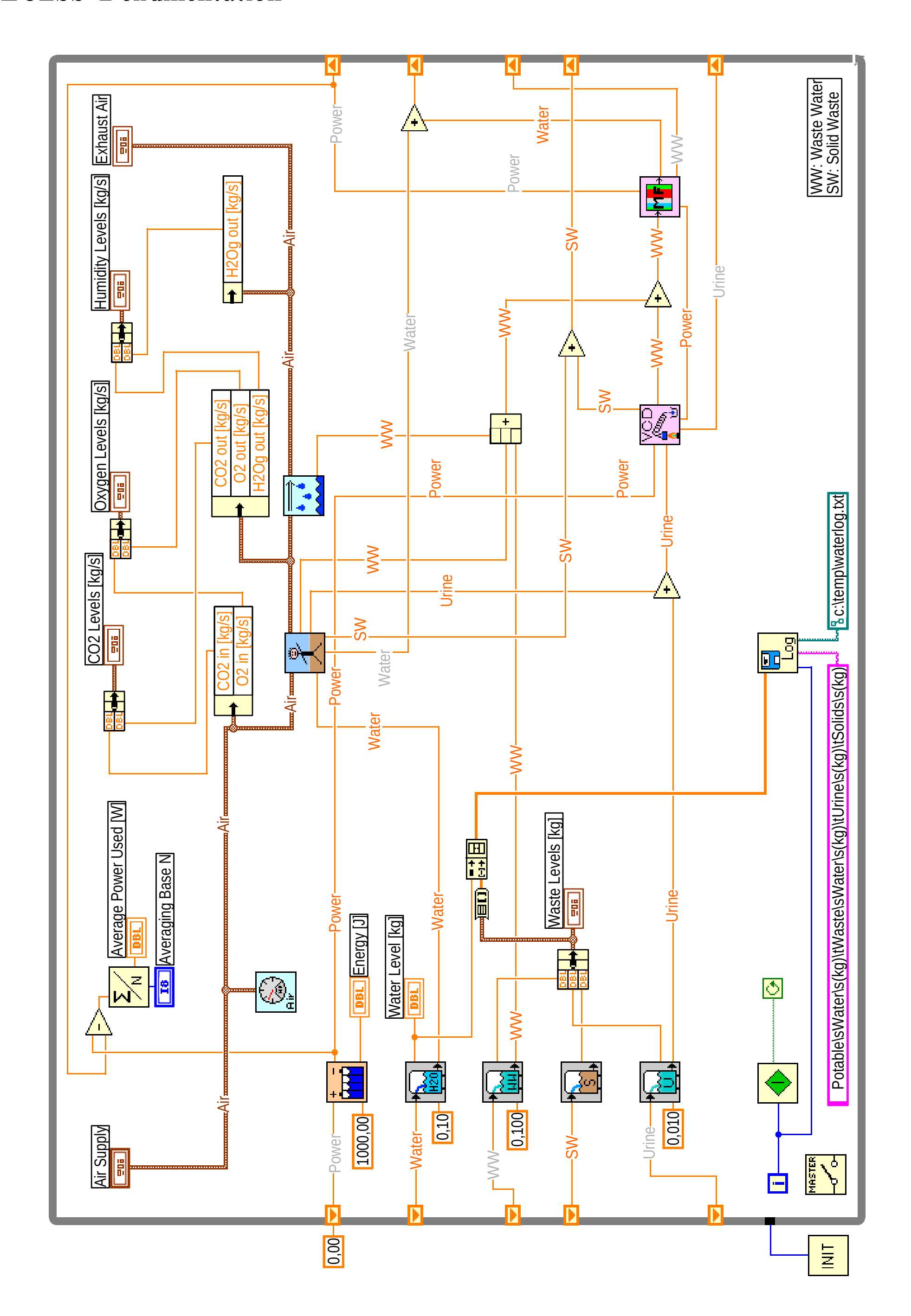
D.1. Example Simulation VI




Power
Waste Water
Water
Multifiltration Output [kg/s]
Multifiltration 4.vi
Water Vapor [kg/s]
Air out
H2Og out [kg/s]
Air
CHX x.vi
Crew s.vi
Air in
Power in [W]
0,00
Power Storage S.vi
Power Demand [W]
1000,00
Waste Water Tank [kg]
Tank Waste Water S.vi
Flow Demand [kg/s]
0,10
Water Tank [kg]
Tank H2O S.vi
WW
Master Control Panel.vi
Simulation Control.vi
Power [W]
Waste Water out [kg/s]
Potable Water [kg/s]
Init.vi
INIT

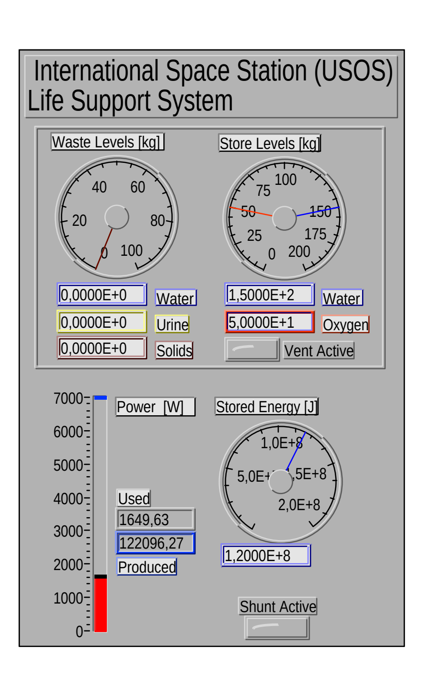
D.2. Closed Water Loop Simulation VI

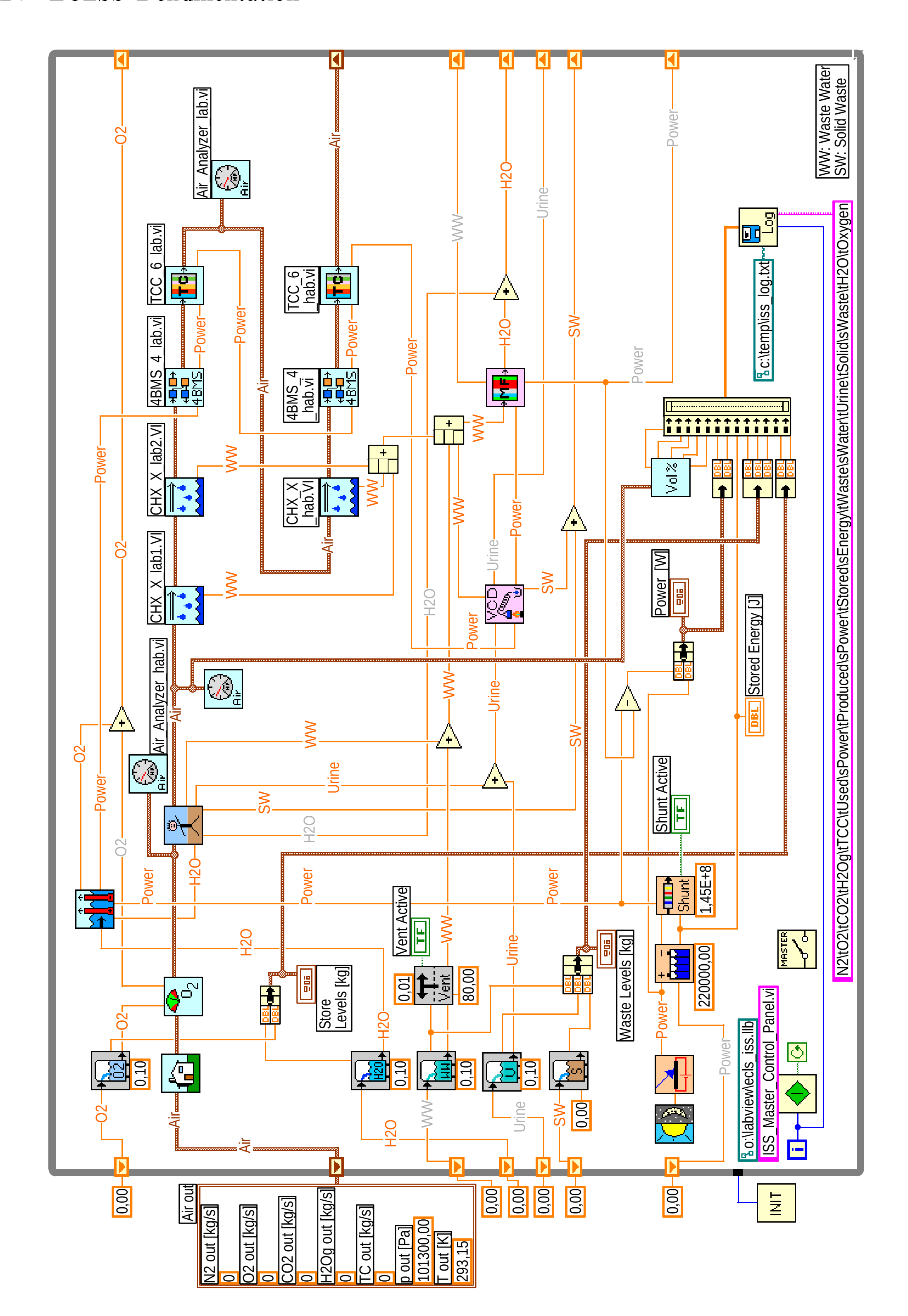




Exhaust Air
Humidity Levels [kg/s]
Oxygen Levels [kg/s]
CO2 Levels [kg/s]
Air Supply
Average Power Used [W]
Averaging Base N
H2Og out [kg/s]
CO2 out [kg/s]
O2 out [kg/s]
CO2 in [kg/s]
O2 in [kg/s]
Energy [J]
Water Level [kg]
Waste Levels [kg]
WW: Waste Water
SW: Solid Waste
c:\temp\waterlog.txt
Potable\sWater\s(kg)\tWaste\s(kg)\tUrine\s(kg)\tSolids\s(kg)
1000,00
0,10
0,100
0,010
0,00
INIT

D.3. ISS ECLSS Simulation VI





WW: Waste Water
SW: Solid Waste
Air_Analyzer_lab.vi
TCC_6_lab.vi
4BMS_4_lab.vi
CHX_X_lab2.VI
CHX_X_lab1.VI
Air_Analyzer_hab.vi
TCC_6_hab.vi
4BMS_4_hab.vi
CHX_X_hab.VI
O2
Air
H2O
WW
Urine
SW
Power
VCD
Vol %
Power [W]
Stored Energy [J]
Shunt Active
Shunt
1,45E+8
220000,00
Vent Active
Vent
0,01
80,00
Store Levels [kg]
Waste Levels [kg]
0,10
0,00
c:\temp\iss_log.txt
Log
N2\tO2\tCO2\tH2Og\tTCC\tUsed\sPower\tProduced\sPower\tStored\sEnergy\tWaste\sWater\tUrine\tSolid\sWaste\tH2O\tOxygen
o:\labview\ecls_iss.llb
ISS_Master_Control_Panel.vi
MASTER
INIT
Air out
N2 out [kg/s]
O2 out [kg/s]
CO2 out [kg/s]
H2Og out [kg/s]
TC out [kg/s]
p out [Pa]
101300,00
T out [K]
293,15

E. MELISSA User Manual

Nachfolgend ist das MELISSA-Benutzerhandbuch eingefügt. Aufgrund des internationalen Nutzerkreises von SSDW und MELISSA wurde es in englischer Sprache verfaßt. Das Handbuch gibt wichtige Hinweise für Benutzer, die eigene Simulationen oder Module erstellen wollen. Auch einige im Hauptteil dieser Diplomarbeit angesprochenen Punkte, z. B. zu den Besonderheiten der Simulationsumgebung, werden vertieft dargestellt.

Damit das Handbuch bei Bedarf direkt aus diesem Anhang kopiert und eigenständig verwendet werden kann, wurde die ursprüngliche Seitennumerierung (von 1 bis 58) beibehalten. Die Seitennummern auf den nächsten 58 Seiten sind also nicht an die Zählung der Diplomarbeit angepaßt.

F. Melissa Kurzreferenz

Auf den folgenden Seiten ist abschließend eine Kurzreferenz für den fortgeschrittenen MELISSA-Anwender eingefügt. Sie faßt auf sieben Seiten die wichtigsten Abbildungen, Tabellen und Grundsätze aus dem MELISSA User Manual (siehe Anhang E ab Seite 247) zusammen.